

ESCUELA POLITÉCNICA DEL EJÉRCITO

**DEPARTAMENTO DE ELÉCTRICA Y
ELECTRÓNICA**

**CARRERA DE INGENIERÍA ELECTRÓNICA Y
TELECOMUNICACIONES**

**PROYECTO DE GRADO PARA LA OBTENCIÓN
DEL TÍTULO DE INGENIERÍA**

**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO
ELECTRÓNICO PARA NAVEGACIÓN GPS BASADO EN
EL DESARROLLO DE UN FORMATO DE ARCHIVO DE
MAPA DIGITAL CAPAZ DE SER MANEJADO POR UN
MICROCONTROLADOR**

PAUL ENRIQUE NICOLALDE AVILA

SANGOLQUÍ - ECUADOR

2011

CERTIFICACIÓN

Certificamos que el presente proyecto de grado titulado “Diseño e Implementación de un Prototipo Electrónico Para Navegación GPS Basado en el Desarrollo de un Formato de Archivo de Mapa Digital Capaz de ser Manejado por un Microcontrolador” ha sido desarrollado en su totalidad, por el señor Paúl Enrique Nicolalde Ávila con C.I. 171500592-0, bajo nuestra dirección.

Ing. Jaime Andrango

DIRECTOR

Ing. Julio Larco

CODIRECTOR

RESUMEN DEL PROYECTO

RESUMEN

En el presente proyecto se desarrolla e implementa un prototipo electrónico portátil que funciona como un navegador GPS por medio de una antena-receptor; donde la información de posicionamiento es relacionada con un mapa digital geo-referenciado el cual es leído desde una memoria flash SD (Secure Digital), y mostrado a través de una pantalla LCD de gráficos. El prototipo está basado en el microcontrolador PIC18F4550. Para la realización del prototipo se desarrolló un nuevo sistema de archivos para gestionar la información del mapa de navegación residente en memoria SD. Se realizaron pruebas del dispositivo utilizando un nuevo formato de archivo de mapa vectorial geo-referenciado desarrollado también en este proyecto.

ABSTRACT

In this project develops and implements a prototype electronic device that works as a GPS navigator via an antenna-receiver, where the positioning information is related to a geo-referenced digital map which is read from a flash memory SD card (Secure Digital), and displayed via an LCD graphics screen. The prototype is based on the PIC18F4550 microcontroller. For the realization of the prototype, a new file system was developed in order to manage information of the map resident in SD memory. The tests of the device was using a new file format for geo-referenced vector map also developed in this project.

DEDICATORIA

Este proyecto está dedicado a mi familia: Padre, Hermanos y en especial a mi Madre que me ha brindado todo su apoyo y sacrificio. A mis amigos y particularmente a mi esposa Katerine por su amor, comprensión y cariño incondicional.

AGRADECIMIENTO

El más sincero agradecimiento a todo el personal docente de la Escuela Politécnica del Ejército, y en especial a los ingenieros: Jaime Andrango y Julio Larco por su valiosa colaboración en la dirección y desarrollo de este trabajo.

PRÓLOGO

El prototipo electrónico desarrollado en este proyecto se lo ha denominado GNOM-4550 que viene del acrónimo inglés *Global Navigation Over Microcontroller* (Navegación Global Sobre Microcontrolador); y 4550 porque está basado en el microcontrolador PIC18F4550, que es el núcleo del sistema. La intención del proyecto no fue la de crear un dispositivo que entre en competencia en el mercado con los navegadores existentes; sino aportar con la investigación en el desarrollo electrónico y de software.

Para el máximo aprovechamiento de los recursos de hardware (y a pesar de que no es el objetivo planteado en el escrito), en la implementación del prototipo se pretende obtener un hardware genérico para aplicaciones múltiples con microcontrolador, donde la diferencia entre una aplicación y otra sea solo el software. Según esta visión, el prototipo electrónico obtenido constituye también una tarjeta de desarrollo para el microcontrolador PIC18F4550.

En dicho prototipo se incluyen periféricos de entrada y salida como los siguientes: pantalla LCD de gráficos de 128 x 64 pixeles con intensidad de luz de fondo (*backlight*) controlada por software, *socket* para tarjeta de memoria SD que permite lectura/escritura, teclado de 6 botones formado por un cursor mas dos botones de selección, puerto serial bidireccional asíncrono RS232, entrada dedicada hacia el conversor A/D por medio de un conector, medición de carga de batería por software, y adaptador de poder externo con opción de recarga de baterías con nivel de corriente constante regulable manualmente.

Por medio de dicha tarjeta de desarrollo se pueden realizar varias aplicaciones con el microcontrolador PIC18F4550 solamente limitado por la complejidad y capacidad de memoria de programa. Por ejemplo se podría realizar un osciloscopio / voltímetro digital, analizador de armónicos de línea eléctrica, trazador de altura de terreno 3D, GPS *data logger*, consola de juegos, etc.

Sobre este dispositivo se realizó también el desarrollo de un analizador de línea eléctrica monofásica que utiliza el canal analógico de entrada con divisor de tensión para conectar directamente hacia los 110V de la red eléctrica. Se programó un osciloscopio en la pantalla LCD gráfica con medición de voltajes RMS y pico, captura de pantalla, escala de tiempo variable, analizador espectral de 0 a 500Hz mediante el algoritmo *Goertzel* y analizador de nivel de armónicos. Todo esto implementado sobre microcontrolador PIC18F4550 corriendo a 48Mhz. Dicho desarrollo se expuso en la Casa Abierta de Proyectos de Innovación y Desarrollo Empresarial (ESPE) 2009.

Durante el desarrollo del presente proyecto se desprendieron varios documentos y aplicaciones importantes, donde algunos de estos, no se plantearon inicialmente. Por ejemplo en la fase inicial, se pensaba manejar la memoria SD por medio del sistema de archivos FAT (*File Allocation Table*) para hacerlo compatible con las PC; sin embargo la documentación era escasa hasta el momento de la investigación y la utilización de librerías ya desarrolladas reducían notablemente la capacidad de memoria de programa del microcontrolador. Por esta razón se optó por desarrollar desde cero un nuevo sistema de archivos para memoria SD (*Secure Digital*). Así nació el sistema FDZ (*File Descriptor Zone*), que constituye un nuevo sistema de archivos creado para gestionar la memoria SD desde un microcontrolador por medio del bus SPI, trabajando directamente sobre la capa física del sistema. Utilizando esta modalidad se logra obtener cierta protección de derechos de utilización del sistema al no tener compatibilidad con FAT. La información grabada en la memoria SD con el sistema FDZ, no se puede visualizar en ninguna computadora personal.

Se desarrollaron a la par también librerías de control para el PIC18F4550, como son: controlador de memoria SD, gestión de archivos, transferencia de archivos por protocolo Xmodem, control de navegación e interpretación de datos GPS, entre otros.

En el proyecto se planteó la creación de un nuevo formato de archivo que sea compatible con microcontrolador y sirva para contener información de mapas digitales para navegación GPS. Se creó entonces el archivo GMV cuyas siglas significan: *Geo-referenced Microcontroller Vector map*, ó mapa vectorial geo-referenciado para microcontrolador. El archivo GMV fue desarrollado también desde cero, definiéndolo de tal manera que su estructura permanezca simple, como para ser interpretada por un microcontrolador que posee una limitada capacidad computacional. Sin embargo el archivo GMV tiene la capacidad de encapsular una gran cantidad de datos acerca de mapas digitales geo-referenciados; limitados principalmente por el hardware disponible.

Casi al culminar el presente proyecto, fue necesario realizar el desarrollo de una herramienta de software para poder construir los mapas para la unidad de manera más rápida; pues, el construir un archivo byte por byte en un editor hexadecimal, resulta un proceso muy lento y complejo. De esta manera en Visual Basic se diseñó el aplicativo de compilador de mapas GMV a partir de archivos KML, contruidos en el entorno gráfico de *GoogleEarth*. El compilador toma los archivos exportados desde *GoogleEarth*, los procesa y construye los archivos de salida para la unidad GNOM-4550.

Finalmente, en el presente documento se pretende explicar y dar la información necesaria de cómo está diseñado e implementado el prototipo, dando a conocer los conocimientos previos, criterios y conceptos que se utilizaron en el desarrollo. No se incluye el código de programación en si; debido a que el texto resultaría demasiado extenso; además por razones de protección de derechos de autor.

ÍNDICE DE CONTENIDO

CAPÍTULO 1: INTRODUCCIÓN	1
1.1 ANTECEDENTES	2
1.2 IMPORTANCIA	2
1.3 ALCANCE	3
1.4 OBJETIVOS	3
1.4.1 General	3
1.4.2 Específico	4
CAPÍTULO 2: MARCO TEÓRICO	5
2.1 SISTEMA DE NAVEGACIÓN GLOBAL POR SATÉLITE GPS: HISTORIA, EVOLUCIÓN Y TEORÍA DE FUNCIONAMIENTO.	
2.1.1 Historia de la navegación por satélite	6
2.1.2 Evolución del sistema GPS	
GPS diferencial (DGPS).....	9
Sistemas de aumentación basados en satélites	10
WAAS (Wide Area Augmentation System)	11
EGNOS (European Geostationary Navigation Overlay Service)...	11
MSAS (Multi-functional Satellite Augmentation System)	11
GAGAN (GPS and GEO Augmented Navigation).....	11
GPS III.....	12
2.1.3 Funcionamiento del sistema GPS.....	13
Segmento espacial.....	13
Segmento de control	13
Segmento de usuario	14
Concepto básico de posicionamiento.....	14
Distancia de usuario y transmisión de código	16
C/A (coarse/acquisition code)	17
P (precision code)	17
Mensaje de navegación	18
2.2 RECEPTORES GPS DE APLICACIÓN GENÉRICA Y PROTOCOLOS DE COMUNICACIÓN UTILIZADOS	
2.2.1 Introducción	20
Efemérides	21
Almanaque	21
WGS84.....	22

2.2.2 El receptor GPS GlobalSat BR-355	22
Interfaz eléctrica	25
Formato de sentencia.....	25
Análisis de trama de datos GPRMC.....	26
Hora UTC	26
Estado	26
Latitud	27
Longitud	27
Velocidad sobre fondo.....	27
Track made good	28
Fecha	28
Variación magnética.....	28
Suma de verificación	29
 2.3 MAPAS DIGITALES APLICADOS A LA NAVEGACIÓN GPS	
2.3.1 Introducción	29
2.3.2 Tipos de mapas digitales	29
2.3.3 Mapas digitales aplicados a la navegación GPS	30
 2.4 ESTRUCTURA Y FUNCIONAMIENTO DE LAS MEMORIAS SD	
2.4.1 Introducción	31
2.4.2 Estructura interna.....	32
2.4.3 Arquitectura de la memoria	32
2.4.4 Características principales.....	33
2.4.5 Registros principales.....	35
Registro de condiciones de operación (OCR)	35
Registro de identificación (CID).....	35
Registro de datos específicos (CSD)	36
Registro de direcciones relativas (RCA).....	36
2.4.6 Interfaz eléctrica.....	36
Configuración de pines.....	36
2.4.7 Topología de bus	37
Bus SD	37
Bus PSI	38
2.4.8 Características eléctricas	38
Encendido	38
Condiciones de operación de bus	40
Niveles de señal	40
2.4.9 Protocolo de comunicación SPI	41
Selección de modalidad	42
Comandos principales.....	43
Tokens de respuesta y tipos de formato	44
Formato R1	44
Formato R1b.....	45

Formato R2.....	45
Formato R3.....	46
Token de respuesta de datos	46
Token de datos	47
Token de error de datos	48
2.4.10 Características de transferencia de datos.....	49
2.4.11 Nuevas tecnologías en tarjetas SD	49
2.5 PANTALLA LCD DE GRÁFICOS	50
2.6 MICROCONTROLADORES: RECURSOS DE COMUNICACIÓN Y HERRAMIENTAS DE DESARROLLO	52
2.6.1 Microcontrolador PIC18F4550	53
2.6.2 Comunicación serial RS-232.....	54
Características básicas	55
Estructura de señal	56
Adaptación de niveles	57
2.6.3 Comunicación serial SPI	57
Bus SPI en el microcontrolador PIC18F4550	59
Transmisión de datos	60
Recepción de datos.....	60
Temporización.....	61
2.6.4 Protocolo de transmisión de datos Xmodem.....	62
2.6.5 Herramientas de desarrollo.....	64
CAPÍTULO 3: DISEÑO DE FORMATO DE ARCHIVO Y SISTEMA DE ARCHIVOS	66
3.1 DISEÑO DEL NUEVO FORMATO DE ARCHIVO DE MAPA VECTORIAL	
3.1.1 Introducción	67
3.2.1 Características generales	67
3.1.3 Definición y nombre de archivo.....	68
3.1.4 Tipo de datos	72
Determinación de una “Zona”	73
Definición de comandos	73
Comando 0xFA Trazado de línea simple.....	74
Comando 0xFB Trazado de línea continua	74
Comando 0xFC Trazado rectangular.....	75
Comando 0xFD Trazado circular	76
Comando 0xFE Colocar texto.....	76
Comando 0xFF Fin de comando / no operación.....	77

3.1.5 Estructura interna del archivo GMV	77
Preámbulo	78
Encabezado	79
Cuerpo del archivo	81
3.2 DISEÑO DEL NUEVO SISTEMA DE ARCHIVOS PARA MEMORIA SECURE DIGITAL (SD)	
3.2.1 Introducción	82
3.2.2 Diseño del sistema de archivos	83
Consideraciones preliminares de la capa física	84
3.2.3 Estructura general del sistema de archivos	85
Direccionamiento.....	86
Unidades	86
Tabla de organización de raíz (ROT)	87
Zona de descriptor de archivos (FDZ)	88
Estructura de la tabla FDT	89
Estructura de la tabla TOP	90
Área de Datos de Usuario (<i>User Data</i>).....	90
3.3 GESTIÓN DEL SISTEMA	
3.3.1 Comunicación con el host.....	92
3.3.2 Formateo de disco	93
3.3.3 Almacenamiento de un archivo.....	94
3.3.4 Lectura de un archivo	96
3.3.5 Eliminación de un archivo	97
3.4 RESUMEN DE CARACTERÍSTICAS	97
CAPÍTULO 4: DISEÑO DEL SOFTWARE DE CONTROL Y PROCESAMIENTO	99
4.1 ESTRUCTURA DEL SOFTWARE DEL SISTEMA	100
4.1.1 Descripción de grupos funcionales de programa	101
4.2 CONTROL DE SISTEMA	102
4.2.1 Configuración básica de microcontrolador	103
4.2.2 Control de interrupciones	103
4.2.3 Memoria RAM y registros principales	104
4.2.4 Ventanas de aviso	105
4.2.5 Funciones gráficas menores	105
4.2.6 Explorador de disco	106

4.3 CONTROL DE CAPA FÍSICA DE DISCO	107
4.3.1 Secuencia de inicialización	108
4.3.2 Lectura de bloque	110
4.3.3 Escritura de bloque	112
4.4 CONTROL DE FORMATO DE DISCO.....	113
4.4.1 Chequeo de formato de disco	114
4.4.2 Formateo de disco	114
4.5 CONTROL DE ENERGÍA.....	115
4.5.1 Nivel de brillo de pantalla	116
4.5.2 Nivel de batería.....	116
Circuito de acoplamiento f(Vbat)	118
Conversor A/D g(Vdiv)	119
Relación voltaje h(N)	120
Relación nivel i(V)	121
4.6 CONTROL DE TRANSFERENCIA DE ARCHIVOS	122
4.6.1 Control de protocolo Xmodem	123
4.6.2 Transferencia y almacenamiento de información.....	123
receive_xmodem_pack (función).....	124
write_data (función).....	125
4.6.3 Registro de archivo en disco.....	126
4.7 CONTROL DE NAVEGACIÓN	126
4.7.1 Núcleo de navegación	127
4.7.2 Acceso a archivos (prelectura).....	129
Petición de archivo	130
Interpretación de nombre de archivo.....	130
Búsqueda de archivo.....	130
Lectura de encabezado y construcción de tabla NFDI	131
4.7.3 Procesamiento de archivos (lectura).....	133
Interpretación de tablas NFDI	133
Lectura de cuerpo de archivo	134
Graficación	134
4.7.4 Tratamiento de datos GPS.....	136
Captura de trama GPS.....	137
Procesamiento de trama de información.....	138
CAPÍTULO 5: DISEÑO Y CONFIGURACIÓN DE HARDWARE	141
5.1 ESTRUCTURA GENERAL DEL SISTEMA	142

5.2 NÚCLEO DEL SISTEMA (MICROCONTROLADOR PIC18F4550).....	142
5.2.1 Estructura y Configuración del Reloj.....	143
5.3 PERIFÉRICOS DE ENTRADA / SALIDA	145
5.3.1 Configuración del puerto de display GLCD	145
5.3.2 Configuración del puerto serial	147
5.3.3 Conexión GPS por puerto serial	149
5.3.4 Configuración del puerto de la tarjeta SD	150
5.3.5 Configuración del puerto de teclado	153
5.3.6 Fuente de poder.....	154
5.3.7 Circuito regulador de corriente constante	155
5.3.8 Circuito regulador de voltaje constante.....	156
5.3.9 Circuito detector	156
5.3.10 Circuito de acoplamiento para medición de carga de batería	157
CAPÍTULO 6: IMPLEMENTACIÓN Y PRUEBAS	158
6.1 IMPLEMENTACIÓN DEL HARDWARE	159
6.1.1 Diseño de placa de circuito impreso	159
6.1.2 Preparación de las placas de circuito impreso.....	161
6.1.3 Montaje de los componentes	161
6.1.4 Ensamblaje sobre material protector	162
6.1.5 Detalle de costos de fabricación	164
6.2 GENERACIÓN DE ARCHIVOS DE PRUEBA	164
6.2.1 Consideraciones iniciales.....	164
6.2.2 Generación de archivo de inicialización S0000E00000.GMV	166
Encabezado - Preámbulo.....	167
Encabezado – Tabla de punteros de bloque (BP).....	168
Cuerpo de archivo	169
6.2.3 Generación de mapa de la zona de prueba	171
Compilador de archivos GMV	171
Construcción de elementos en <i>GoogleEarth</i>	172
Compilación de archivos GMV a partir de archivos KML	174
6.3 ALMACENAMIENTO DE ARCHIVOS EN MEDIO FÍSICO.....	177
6.4 PRUEBAS FINALES DE LA UNIDAD GNOM-4550	180
6.4.1 Introducción	180
6.4.2 Primera prueba: archivo de inicialización.....	180
6.4.3 Segunda prueba: prueba con receptor GPS	182

ANEXO 1 DIAGRAMA ELECTRÓNICO (Tarjeta principal)	187
ANEXO 2 DIAGRAMA ELECTRÓNICO (Fuente de poder)	189
ANEXO 3 DISEÑO DE CIRCUITO IMPRESO	191
ANEXO 4 RESUMEN DE PRICIPALES FUNCIONES DE LIBRERÍA.....	193
ANEXO 5 MENSAJES DEL SISTEMA (Ventanas de aviso).....	202
ANEXO 6 ENTRADAS / SALIDAS DEL MICROCONTROLADOR.....	204
ANEXO 7 CARACTERÍSTICAS DEL SISTEMA DE ARCHIVOS FDZ	206
ANEXO 8 HOJAS TÉCNICAS (<i>Datasheets</i>)	208
REFERENCIAS BIBLIOGRÁFICAS	222

ÍNDICE DE TABLAS

Tabla 2.1 Resumen de especificaciones del receptor GPS GlobalSat BR-355.....	23
Tabla 2.2 Características física de los tipos de memoria flash SD.....	31
Tabla 2.3 Registros principales de la tarjeta SD.....	35
Tabla 2.4 Configuración de terminales para el modo SD	37
Tabla 2.5 Configuración de terminales para el modo SPI	37
Tabla 2.6 Condiciones de operación de bus SD	40
Tabla 2.7 Niveles de voltaje en el bus SD.....	41
Tabla 2.8 Resumen de comandos principales de tarjeta SD.....	43
Tabla 2.9 Comparación de tasas de datos de tarjetas MMC y SD	49
Tabla 2.10 Distribución de pines del GLCD TS12864B.....	52
Tabla 2.11 Principales características del microcontrolador PIC18F4550	54
Tabla 2.12 Descripción de pines en los conectores usados en RS232.....	56
Tabla 2.13 Caracteres especiales en el protocolo Xmodem	62
Tabla 3.1 Relación entre dirección lógica y dirección física	86
Tabla 3.2 Resumen de características del sistema de archivos FDZ.....	98
Tabla 4.1 Elementos de memoria RAM y sus funciones	105
Tabla 4.2 Información en caché de proceso para la función de inicialización	110
Tabla 4.3 Contenido de caché de proceso para la función “ <i>read_block</i> ”	111
Tabla 4.4 Contenido de caché de proceso para la función “ <i>write_block</i> ”	113
Tabla 5.1 Detalle de conexión de puerto del display GLCD	147
Tabla 5.2 Detalle de conexión del puerto serial RS-232	148

Tabla 5.3 Comparación de conexiones PS/2 entre GPS y PC normalizado	149
Tabla 5.4 Valores de niveles de voltaje en el bus de la tarjeta SD.....	151
Tabla 5.5 Valores de consumo de corriente de la memoria SD	152
Tabla 5.6 Detalle de conexión de puerto de la tarjeta SD	153
Tabla 5.7 Detalle de conexión de puerto de teclado	154
Tabla 6.1 Costo directo de implementación del prototipo.....	164

ÍNDICE DE FIGURAS

Figura 2.1 Constelación TRANSIT	7
Figura 2.2 Satélite Navstar-2F.....	13
Figura 2.3 Posición de usuario en el caso unidimensional	14
Figura 2.4 Posición de usuario en el caso bidimensional.....	15
Figura 2.5 Ejemplo de modulación sobre la portadora L1	18
Figura 2.6 Formato básico del mensaje de navegación GPS.....	18
Figura 2.7 Diagrama de bloques del receptor GPS.....	20
Figura 2.8 Receptor GPS GlobalSat BR-355 con conector PS/2	22
Figura 2.9 Gráfica de explicación de una milla náutica	27
Figura 2.10 Ejemplo de vector “Track made good ó Track – True”	28
Figura 2.11 Ejemplo de un mapa digital vectorizado.....	30
Figura 2.12 Estructura básica de la memoria SD	32
Figura 2.13 Arquitectura de la memoria SD	33
Figura 2.14 Topología de bus SPI.....	39
Figura 2.15 Diagrama de encendido (power up)	39
Figura 2.16 Gráfico de niveles de voltaje en el bus de la tarjeta SD	41
Figura 2.17 Estructura de los comandos SPI	42
Figura 2.18 Estructura de respuesta R1.....	44
Figura 2.19 Estructura de respuesta R2.....	45
Figura 2.20 Estructura de respuesta R3.....	46
Figura 2.21 Estructura de byte de respuesta de datos.....	46

Figura 2.22 Estructura de token de datos	47
Figura 2.23 Estructura de token de datos 2	48
Figura 2.24 Estructura de token de datos 3	48
Figura 2.25 Estructura de token de error de datos	48
Figura 2.26 Tarjeta de memoria SD con tecnología inalámbrica Eye-Fi	49
Figura 2.27 Vista frontal del GLCD TS12864B.....	50
Figura 2.28 Distribución de memoria en la pantalla GLCD	51
Figura 2.29 Diagrama de bloques de la estructura interna del GLCD	51
Figura 2.30 Ejemplo de microcontrolador: MC68HC11 de Motorola	53
Figura 2.31 Conexión RS232 asíncrona	56
Figura 2.32 Estructura de la señal RS232.....	57
Figura 2.33 Ejemplo de configuración de bus SPI.....	58
Figura 2.34 Diagrama de conexión Master/Slave de bus SPI	59
Figura 2.35 Diagrama de bloques interno del controlador SPI.....	60
Figura 2.36 Diagrama de temporización del bus SPI	61
Figura 2.37 Secuencia de transmisión de datos en Xmodem	64
Figura 3.1 Ejes coordenados del mapa de la tierra y sus límites	69
Figura 3.2 División en zonas del archivo GMV	70
Figura 3.3 Ejemplo que explica el significado del nombre de archivo GMV	70
Figura 3.4 Ejemplo de ordenamiento de archivos GMV	71
Figura 3.5 Delimitación entre zonas del archivo GMV.....	71
Figura 3.6 Rango de datos que diferencian coordenadas y comandos.....	72
Figura 3.7 Operación gráfica del controlador en una “Zona”	73
Figura 3.8 Estructura de un comando	74
Figura 3.9 Ejemplo del comando 0xFA	74
Figura 3.10 Ejemplo de comando 0xFB	75
Figura 3.11 Ejemplo de comando 0xFC	75

Figura 3.12 Ejemplo de comando 0xFD	76
Figura 3.13 Ejemplo de comando 0xFE	76
Figura 3.14 Estructura interna del archivo GMV.....	77
Figura 3.15 Estructura interna del preámbulo del archivo GMV	78
Figura 3.16 Ejemplo de puntero de 16bits definido en el encabezado	80
Figura 3.17 Estructura del encabezado del archivo GMV	80
Figura 3.18 Estructura del cuerpo del archivo GMV	82
Figura 3.19 Persistencia de salto en el direccionamiento del disco SD	84
Figura 3.20 Áreas principales del sistema de archivos FDZ	85
Figura 3.21 Estructura de la tabla ROT	87
Figura 3.22 Ejemplo de indicador de tamaño de disco.....	88
Figura 3.23 Estructura de la zona de descriptor de archivos FDZ.....	88
Figura 3.24 Estructura de un bloque FDB	89
Figura 3.25 Estructura interna del área de datos de usuario.....	90
Figura 3.26 Bloque CSI y unidad de memoria.....	91
Figura 3.27 Comunicación con host por medio de puerto serial.....	92
Figura 3.28 Identificación de archivos GMV en el sistema	93
Figura 3.29 Almacenamiento de un archivo en el disco	96
Figura 3.29 Almacenamiento de un archivo en el disco	96
Figura 3.30 Lectura de un archivo en el disco.....	96
Figura 4.1 Estructura de los módulos que componen el software del sistema....	100
Figura 4.2 Menú principal y grupos funcionales del sistema	101
Figura 4.3 Estructura interna de la memoria RAM del sistema	104
Figura 4.4 Ejemplo de ventana de aviso con mensaje de alerta	105
Figura 4.5 Pantalla de explorador de disco	106
Figura 4.6 Mensaje de confirmación de archivo a borrar	107
Figura 4.7 Pantalla de error de sistema con información de caché de proceso ..	108

Figura 4.8 Diagrama de flujo simplificado de la función “ <i>init_sequence</i> ”	109
Figura 4.9 Procedimiento de lectura de bloque en disco SD.....	110
Figura 4.10 Procedimiento de escritura de bloque en disco SD.....	112
Figura 4.11 Pantalla de tipos de formateo de disco	114
Figura 4.12 Contenido en ROM de tabla ROT	115
Figura 4.13 Pantalla de control de brillo de GLCD	116
Figura 4.14 Etapas que atraviesa la señal de carga de batería en el sistema ...	117
Figura 4.15 Relación entre señales de entrada y salida del circuito de acoplamiento	118
Figura 4.16 Relación entre las señales de entrada y salida del conversor A/D...	120
Figura 4.17 Relación entre el voltaje de batería y el nivel de carga	121
Figura 4.18 Diagrama de flujo simplificado de la función “ <i>receive_external_file</i> ”	123
Figura 4.19 Diagrama de flujo simplificado de la función “ <i>receive_xmodem_pack</i> ”	124
Figura 4.20 Diagrama simplificado de la función “ <i>write_data</i> ”	125
Figura 4.21 Diagrama de flujo simplificado de la función “ <i>make_file</i> ”	126
Figura 4.22 Cuadrantes del control de navegación y área de pantalla LCD	127
Figura 4.23 Inicialización del núcleo de navegación y el registro NAVREG	128
Figura 4.24 Interpretación de nombres de archivo	130
Figura 4.25 Estructura de la memoria caché formada por cuatro tablas NFDI....	131
Figura 4.26 Ejemplo de interpretación de tabla NFDI.....	133
Figura 4.27 Pantalla de configuración de flags en la unidad GNOM-4550	134
Figura 4.28 Estructura de la función de graficación	135
Figura 4.29 Ejemplo de interpretación de datos del cuerpo de archivo.....	136
Figura 4.30 Ejemplo de trazo de líneas y texto en la unidad GNOM-4550.....	136
Figura 4.31 Diagrama de identificación de trama GPS	138
Figura 4.32 Diagrama de procesamiento de trama de información.....	139

Figura 4.33 Desensamble de trama GPRMC en las variables del registro GPSREG	140
Figura 5.1 Estructura en bloques del sistema	142
Figura 5.2 Estructura del reloj interno del PIC18F4550.....	143
Figura 5.3 Configuración del PLL interno del microcontrolador.....	144
Figura 5.4 Diagrama de conexiones del display GLCD.....	146
Figura 5.5 Circuito de acoplamiento RS232 / TTL.....	147
Figura 5.6 Comparación de señales en el circuito de acoplamiento RS232 / TTL	148
Figura 5.7 Conectores PS/2 macho y hembra.....	149
Figura 5.8 Adaptador DB-9 / USB a GPS PS/2	150
Figura 5.9 Gráfico de niveles de voltaje en el bus de la tarjeta SD	151
Figura 5.10 Circuito de acoplamiento TTL a SD.....	151
Figura 4.21 Diagrama de flujo simplificado de la función “ <i>make_file</i> ”.....	126
Figura 4.22 Cuadrantes del control de navegación y área de pantalla LCD	127
Figura 4.23 Inicialización del núcleo de navegación y el registro NAVREG	128
Figura 4.24 Interpretación de nombres de archivo	130
Figura 4.25 Estructura de la memoria caché formada por cuatro tablas NFDI....	131
Figura 4.26 Ejemplo de interpretación de tabla NFDI.....	133
Figura 4.27 Pantalla de configuración de flags en la unidad GNOM-4550	134
Figura 4.28 Estructura de la función de graficación	135
Figura 4.29 Ejemplo de interpretación de datos del cuerpo de archivo.....	136
Figura 4.30 Ejemplo de trazo de líneas y texto en la unidad GNOM-4550.....	136
Figura 4.31 Diagrama de identificación de trama GPS	138
Figura 4.32 Diagrama de procesamiento de trama de información.....	139
Figura 4.33 Desensamble de trama GPRMC en las variables del registro GPSREG	140
Figura 5.1 Estructura en bloques del sistema	142

Figura 5.2 Estructura del reloj interno del PIC18F4550.....	143
Figura 5.3 Configuración del PLL interno del microcontrolador.....	144
Figura 5.4 Diagrama de conexiones del display GLCD.....	146
Figura 5.5 Circuito de acoplamiento RS232 / TTL.....	147
Figura 5.6 Comparación de señales en el circuito de acoplamiento RS232 / TTL	148
Figura 5.7 Conectores PS/2 macho y hembra.....	149
Figura 5.8 Adaptador DB-9 / USB a GPS PS/2.....	150
Figura 5.9 Gráfico de niveles de voltaje en el bus de la tarjeta SD	151
Figura 5.10 Circuito de acoplamiento TTL a SD.....	151
Figura 5.11 Diagrama de conexión del zócalo de tarjeta SD.....	152
Figura 5.12 Circuito de interfaz de teclado.....	153
Figura 5.13 Diagrama en bloque de la fuente de poder	154
Figura 5.14 LM317 como fuente de corriente constante	155
Figura 5.15 Diagrama simplificado del circuito detector en la fuente	156
Figura 5.16 Diagrama del circuito de acoplamiento para medición de carga de batería	157
Figura 6.1 Simulación 3D de la placa principal en capa superior	160
Figura 6.2 Simulación 3D de la placa principal en capa inferior	160
Figura 6.3 Cara superior de la placa madre obtenida por fotograbado	161
Figura 6.4 Vista superior de la placa madre con componentes montados	162
Figura 6.5 Ilustración del prototipo ensamblado.....	163
Figura 6.6 Presentación del prototipo GNOM-4550	163
Figura 6.7 Creación de un nuevo archivo en WinHex	167
Figura 6.8 Creación de preámbulo en WinHex.....	167
Figura 6.9 Creación de tabla de punteros de bloque.....	168
Figura 6.10 Gráfica de delimitación de zonas de archivo de inicialización.....	169
Figura 6.11 Separación entre bloques y relación en bytes del archivo de inicialización	170

Figura 6.12 Ejemplo de código hexadecimal en bloque de archivo de inicialización	170
Figura 6.13 Ejemplo de archivo KML del cual se extraen las coordenadas de puntos de objeto geo-referenciado	172
Figura 6.14 Creación de elementos de mapa en <i>GoogleEarth</i> a partir de herramientas de ruta, polígono y punto	173
Figura 6.15 Exportación de elementos de mapa como archivos KML en <i>GoogleEarth</i>	173
Figura 6.16 Pantalla de programa compilador de archivos GMV	174
Figura 6.17 Archivos GMV de salida del compilador	175
Figura 6.18 Intersección de línea con bordes de zona de un archivo GMV	176
Figura 6.19 Memoria SD en zócalo de la unidad GNOM-4550	177
Figura 6.20 Pantalla de configuración de conexión nueva en <i>Hyperterminal</i>	178
Figura 6.21 Pantalla de envío de archivo en <i>Hyperterminal</i>	178
Figura 6.22 Pantalla de recepción de paquetes en el prototipo	179
Figura 6.23 Pantalla de explorador en prototipo con el archivo recibido	179
Figura 6.24 Menú principal de la unidad GNOM-4550	180
Figura 6.25 Pantalla de inicialización de receptor GPS	181
Figura 6.26 Pantalla de inicialización de receptor GPS (opciones)	181
Figura 6.27 Despliegue de información del archivo de inicialización	182
Figura 6.28 Menú principal de la unidad GNOM-4550	183
Figura 6.29 Pantalla de inicialización de receptor GPS	183
Figura 6.30 Pantallas de los modos Manual y GPS de la unidad GNOM-4550...	184
Figura 6.31 Ejemplo de navegación GPS en área de prueba con archivos generados	184

ÍNDICE DE HOJAS TÉCNICAS

Especificación de baterías GP modelo GP17R8H	209
Microchip PIC18F4550	210
Maxim MAX232	213
National Semiconductor LM317T	215
National Semiconductor LM2936-3.3	216
ST Semiconductors L7800 Series (LM7805).....	217
TS12864B LCD gráfico	218
GlobalSat RS232 GPS receiver	220

GLOSARIO

3D . -	Referente a tres dimensiones, ancho alto profundidad ó latitud longitud y altura.
A/D (Analógico / Digital) . -	Referente a conversión Analógico - Digital.
ACK (Acknowledge) . -	Caracter de protocolo Xmodem.
ADC (Analog to Digital Converter) . -	Conversor de Analógico a Digital.
AI (Adobe Illustrator) . -	Extensión de archivo Adobe Illustrator.
Almanac (Almanaque) . -	Conjunto de valores ó parámetros que permiten al GPS predecir la órbita de satélites.
ALPHA . -	Proyecto de posicionamiento hiperbólico de la Ex-Unión Soviética.
ArcGis . -	Software cartografía y sistemas de información geográfica.
ArcView . -	Software de análisis e integración de datos cartográficos.
ASCII (American Standard Code for Information Interchange) . -	Código Estadounidense Estándar para el Intercambio de Información. Código binario estándar asignado a caracteres y números.
Backlight . -	Luz de fondo - utilizado en pantallas LCD.
BMP (Bitmap) . -	Extensión de archivo de mapa de bits.
BP (Block Pointer) . -	Puntero de bloque - Referido al sistema de archivos FDZ.
BYTE . -	Conjunto de 8 bits que se interpretan como una unidad de información digital.
C/A (Coarse / Acquisition code) . -	Código de pseudo-ruido randómico o PRN con el cual se modulan datos de GPS para uso civil.
CAN (Cancel) . -	Caracter de utilización en el protocolo Xmodem.
CCS (Custom Computer Services) . -	Compilador utilizado para programación de microcontroladores de Microchip.
CDR (Corel Draw) . -	Extensión de archivo CorelDraw.
CHAYKA . -	Proyecto de posicionamiento hiperbólico de la Ex-Unión Soviética.
CID (Card Identification) . -	Registro de identificación de tarjeta SD.
CKE (Clock Enable) . -	Bit de configuración de reloj de bus SPI en el microcontrolador PIC18F4550.
CKP (Clock Polarity) . -	Bit de configuración de reloj de bus SPI en el microcontrolador PIC18F4550.
CLK (Clock) . -	Terminal de señal de reloj de un dispositivo.
Cold Start . -	Inicialización en frío. Modo de inicialización de un receptor GPS.
COMPASS . -	Proyecto GPS Chino operativo desde el año 2012.
CR (Carriage Return) . -	Caracter ASCII de retorno de carro.
CRC (Cyclic Redundance Code) . -	Comprobación de Redundancia Cíclica. Código para corrección de errores de datos.
CS (Chip Select) . -	Selección de Chip. Terminal eléctrico de un dispositivo digital.
CSD (Card Specific Data) . -	Registro de datos específicos de tarjeta SD.

CSI (Content Status Indicator) . -	Indicador de Estado de Contenido. Primer byte del primer bloque de cada unidad del sistema de archivos FDZ que indica el estado de ocupación de la misma.
D/I (Data / Instruction) . -	Terminal de control del GLCD.
DAT . -	Terminal de datos de tarjeta SD.
Data Logger (GPS) . -	Dispositivo electrónico que almacena en algún tipo de memoria datos de magnitudes variantes en el tiempo (datos de GPS en este caso).
DC/DC (Direct Current) . -	Convertor de voltaje y/o corriente de DC a DC.
DCE (Data Communication Equipment) . -	Equipo de Comunicación de Datos. Referente a comunicación serial RS232.
DDRAM (Double Data Rate RAM) . -	Memoria RAM de doble tasa de datos. Realiza transacciones tanto en el ciclo de subida del reloj como en el ciclo de bajada.
DEC . -	Decimal.
DECCA . -	Proyecto de posicionamiento hiperbólico del Reino Unido.
DGPS (Differential - GPS) . -	GPS Diferencial.
DIP (Dual In Line Package) . -	Empaque doble en línea para circuitos integrados.
Doppler . -	Efecto de aparente cambio de frecuencia de una onda producido por el movimiento relativo entre la fuente y el emisor.
DPI (Dots per Inch) . -	Puntos por Pulgada - Unidad que expresa el nivel de resolución en una pantalla o documento impreso.
DTE (Data Terminal Equipment) . -	Equipo Terminal de Datos. Referente a comunicación serial RS232.
DWG (Autocad Drawing) . -	Extensión de archivo Autocad.
E/S . -	Entrada / Salida
Efemérides . -	Datos transmitidos del satélite al receptor GPS.
EGNOS (European Geostationary Navigation Overlay Service) . -	Sistema de aumentación basado en satélite.
EIA-422A . -	Estándar para transmisión de datos seriales por bus balanceado.
EOT (End Of Transmission) . -	Fin de la Transmisión. Caracter usado en protocolo Xmodem.
ESA (European Spatial Agency) . -	Agencia Espacial Europea.
EUA . -	Estados Unidos de América.
Eye-Fi . -	Tarjeta SD con capacidad de conexión inalámbrica análoga al Wi-Fi.
FAT (File Allocation Table) . -	Tabla de Asignación de Archivos. Sistema de archivos desarrollado para computadoras personales utilizado en el formato de discos duros, y memorias externas.
FDB (File Descriptor Block) . -	Bloque Descriptor de Archivo. Bloque de información en el área de sistema del disco formateado con FDZ.
FDT (File Description Table) . -	Tabla de Descriptor de Archivo. Parte de un bloque FDB.
FDZ (File Description Zone) . -	Zona de Descriptor de Archivo. Sistema de archivos para memorias SD desarrollado en este proyecto.
GAGAN (GPS and GEO Augmented Navigation) . -	Sistema de aumentación basado en satélite.
GALILEO . -	Sistema de posicionamiento por satélite de la Unión Europea.
GB (GigaByte) . -	Mil millones de bytes.

Geoide . -	Forma de la tierra que adquiere por achatamiento en los polos debido a la acción rotacional y gravitacional de la misma.
Geo-referenciado . -	Información que se encuentra relacionada con una posición geográfica conocida.
GGA . -	Tipo de sentencia de datos de protocolo NMEA0183.
GIF (Graphic Interchange Format) . -	Formato de intercambio de Gráficos, utilizado para gráficos en World Wide Web y animaciones.
GIS (Graphic Information System) . -	Sistema de Información Geográfica. Es la integración de hardware, software diseñada para manipular datos geográficos.
GLCD (Graphic LCD) . -	LCD de gráficos. Usualmente formada por una matriz de puntos ó píxeles.
GLL . -	Tipo de sentencia de datos de protocolo NMEA0183.
GlobalSat . -	Marca registrada de empresa que fabrica receptores GPS.
GLONASS (GLObalnaya NAVigatsionnaya Sputnikovaya Sistema) . -	Sistema de Navegación por Satélite Orbitante.
GMT (Greenwich Mean Time) . -	Tiempo promedio del Observatorio de Greenwich, en Londres. También tiempo UTC.
GMV (Geo-referenced Microcontroller Vector map) . -	Mapa Vectorial Geo-referenciado para Microcontrolador. Nuevo formato de archivo desarrollado en este proyecto.
GND (Ground) . -	Terminal de tierra en un dispositivo ó circuito.
GNOM-4550 (Global Navigation Over Microcontroller) . -	Navegación Global Sobre Microcontrolador. Nombre que se le ha dado al prototipo diseñado y construido en base del microcontrolador PIC18F4550.
Goertzel . -	Algoritmo de procesamiento digital que permite identificar las componentes de frecuencia de una señal.
GPRMC (GPS - Recommended Minimum information) . -	Trama de datos de GPS de información mínima recomendada.
GPSREG (GPS Register) . -	Conjunto de registros internos del sistema operativo que almacenan datos del receptor GPS.
GSA . -	Tipo de sentencia de datos de protocolo NMEA0183.
Handshake . -	Control de flujo para el intercambio de datos seriales RS232.
HC (High Capacity) . -	Referente a la tecnología utilizada en memorias SD de gran capacidad.
HEX . -	Hexadecimal.
Hot Start . -	Inicialización en frío. Modo de inicialización de un receptor GPS.
How (Hand Over Word) . -	Parte de la trama de datos del mensaje de navegación GPS.
I2C (Inter Integrated Circuit) . -	Protocolo de comunicación entre circuitos integrados que residen en una misma placa.
IDLE . -	Estado de espera en que puede encontrarse el sistema interno de una tarjeta SD.
INOCAR . -	Instituto Oceanográfico de la Armada del Ecuador.
ISRO . -	Agencia India del Espacio.
JPG (Joint Photographic Experts Group) . -	Extensión de archivos de gráficos almacenados con determinado algoritmo de compresión.
KN (Knots) . -	Nudos, unidad de velocidad usada en navegación y aviación equivalente a 1852Km / h.
Knots . -	Unidad de velocidad (nudos ó Kn).

LCD (Liquid Crystal Display) . -	Display de cristal líquido - en el caso de gráficos está formada por una matriz de puntos.
LF (Line Feed) . -	Caracter ASCII de avance de línea.
LORAN . -	Proyecto de posicionamiento hiperbólico de los Estados Unidos.
LSB (Less Significant Bit) . -	Bit menos significativo.
MBR (Master Boot Record) . -	Primer sector de un medio digital con información para su arranque y funcionamiento.
MCS (Master Control Station) . -	Estación de Control Principal.
Microcontrolada . -	Referente a Microcontrolador
Microcontrolador . -	Dispositivo electrónico que contiene un sistema computacional completo integrado en un solo chip.
MIPS (Million of Operation per Second) . -	Millones de operaciones por segundo. Unidad de medida relativa de capacidad de procesamiento de un microprocesador.
MMC (Multimedia Card) . -	Tarjeta multimedia de memoria de estado sólido no volátil.
MSAS (Multi-functional Satellite Augmentation System) . -	Sistema de aumentación basado en satélite.
MSB (Most Significant Bit) . -	Bit más significativo.
MSSP (Master Synchronous Serial Port) . -	Puerto maestro serial síncrono. Elemento interno de comunicaciones del microcontrolador.
MUX . -	Multiplexor
NAK (Not - Acknowledge) . -	Caracter de protocolo Xmodem.
NAVREG (Navigation Register) . -	Conjunto de registros internos del sistema operativo que almacenan datos de navegación.
Navstar GPS . -	Nombre completo del sistema GPS.
NFDI (Navigation File Descriptor Indicator) . -	Tabla de punteros en el sistema de archivos FDZ.
Ni-MH (Níquel Metal Hidruro) . -	Tipo de batería de níquel e hidruro metálico.
NMEA (National Marine Electronics Association) . -	Asociación Nacional de Electrónica para la Marina.
NMEA0183 . -	Protocolo utilizado para comunicación de datos entre dispositivos de uso marino (GPS).
OCR (Operating Conditions Register) . -	Registro de condiciones de operación de tarjeta SD.
OMEGA . -	Proyecto de posicionamiento hiperbólico de los Estados Unidos.
P/C (Precision Code) . -	Código de pseudo-ruido randómico con el cual se modulan datos de GPS para uso militar.
PC (Personal Computer) . -	Computadora Personal.
PDF (Portable Document Format) . -	Extensión de archivo Adobe Document Reader.
PIC18F4550 . -	Nombre del microcontrolador utilizado en el presente prototipo de la empresa Microchip.
PLL (Phase Locked Loop) . -	Anillo de enganche de fase - oscilador.
PPS (Precise Positioning Service) . -	Servicio de Posicionamiento Preciso.
PRN (Pseudo Random Noise) . -	Código de pseudo-ruido randómico ó código C/A.

Proprietary Sentences . -	Sentencias propietarias (referido al protocolo NMEA)
PS/2 (IBM Personal System/2) . -	Conector utilizado para periféricos de computador como teclado y mouse.
pseudodistancia . -	Distancia entre el satélite y el receptor en tierra con un valor alto de incertidumbre.
pseudo-ruido . -	Secuencia de datos generados a partir de un algoritmo que se asemejan a una señal aleatoria.
PWM (Pulse Width Modulator) . -	Modulador de ancho de pulso.
R/W (Read / Write) . -	Lectura / Escritura.
RAM (Random Access Memory) . -	Memoria de Acceso Aleatorio.
RASTER . -	Tipo de mapa digital el cual se forma por fotografías.
RCA (Relative Card Address) . -	Registro de direcciones relativas de tarjeta SD.
RDS (Emisora FM) . -	Datos textuales que son añadidos a una señal de radio FM analógica para lograr observarlos en un receptor preparado.
RF (Radio Frequency) . -	Radio Frecuencia.
RMC . -	Tipo de sentencia de datos de protocolo NMEA0183.
RMS (Root Mean Square) . -	Raíz Media Cuadrática es una medida estadística de una magnitud que varía cuantitativamente en el tiempo.
ROT (Root Organization Table) . -	Tabla de Organización de Raíz. Conjunto de datos en el área de sistema de un disco con formato FDZ.
RRAM (Read RAM) . -	Memoria RAM de 512 bytes definida en el sistema operativo del microcontrolador.
RS232 . -	Interfaz estandarizada para transmisión serial de datos digitales entre dos dispositivos.
RXD . -	Terminal de recepción de datos seriales.
SA (Selective Availability) . -	Disponibilidad Selectiva. Grado de error aleatorio introducido en el sistema GPS.
Satellite spoofing . -	Interferencia deliberada de una señal de satélite.
SCK (SPI bus Clock) . -	Reloj de bus SPI.
SD / Flash SD (Secure Digital) . -	Memoria flash de datos digitales no volátil de estado sólido.
SDA (Secure Digital Asociation) . -	Asociación Secure Digital
SDI . -	Terminal de entrada de datos del bus SPI.
SDO . -	Terminal de salida de datos del bus SPI.
SiRF Star . -	Circuitería que incorpora un receptor GPS.
SOH . -	Caracter de protocolo Xmodem.
SOL . -	Servicios de Seguridad Para la Vida.
SPI . -	Terminal de entrada de datos del bus SPI.
SPS (Standard Positioning Service) . -	Servicio de Posicionamiento Estándar.
SS (Spread Spectrum) . -	Espectro Ensanchado.
SSPBUF . -	Byte de estado ó configuración del bus SPI en el microcontrolador.
SSPCON1 . -	Byte de estado ó configuración del bus SPI en el microcontrolador.
SSPSR . -	Byte de estado ó configuración del bus SPI en el microcontrolador.

SSPSTAT . -	Byte de estado ó configuración del bus SPI en el microcontrolador.
TIMATION . -	Experimento Estadounidense de navegación por satélite.
TLM (TeLeMetry) . -	Telemetría.
Token . -	Paquete de datos.
TOP (Table Of unit Pointers) . -	Tabla de punteros a unidad. Parte de un bloque FDB en el sistema de archivos FDZ.
Tow (Time of Week) . -	Parte de la trama de datos del mensaje de navegación GPS.
Track made good . -	Vector de corrección de rumbo por influencia del viento usado en aeronáutica.
TRAM . -	Memoria RAM de 16 bytes definida en el sistema operativo del microcontrolador.
TRANSIT . -	Experimento Estadounidense de navegación por satélite.
TSICADA . -	Experimento de la Ex-Unión Soviética de navegación por satélite.
TTL (Transistor Transistor Logic) . -	Tecnología utilizada en los circuitos integrados.
TXD . -	Terminal de transmisión de datos seriales.
UP (Unit Pointer) . -	Puntero a Unidad. Parte del bloque FDB en un sistema de archivos FDZ.
URSS . -	Unión de Repúblicas Socialistas Soviéticas.
USB (Universal Serial Bus) . -	Comunicación de bus serial entre un computador y un periférico.
UTC (Universal Time Coordinated) . -	También conocido como Greenwich Mean Time es una escala de tiempo mundial con base en el tiempo del meridiano cero.
VDD . -	Terminal de voltaje positivo para alimentación de circuitos digitales.
VEE . -	Terminal del GLCD con voltaje referencial de -20V.
VSS . -	Terminal de voltaje negativo para alimentación de circuitos digitales.
VTG . -	Tipo de sentencia de datos de protocolo NMEA0183.
WAAS (Wide Area Augmentation System) . -	Sistema de aumentación basado en satélite.
Warm Start . -	Inicialización en tibio. Modo de inicialización de un receptor GPS.
WGS84 (World Geodetic System 1984) . -	Sistema de referencia con centro en el centro de masas de la tierra que toma en cuenta la forma de geoide de la misma.
Wi-Fi (Wi-Fi Alliance (WECA)) . -	Marca que prueba, adopta y certifica equipos y elementos de red inalámbricos que cumplen con el estándar IEEE 802.x
WinHex . -	Programa editor hexadecimal bajo Windows.
WRAM (Write RAM) . -	Memoria RAM de 512 bytes definida en el sistema operativo del microcontrolador.
Xmodem . -	Protocolo de transmisión de datos digitales entre dispositivos por medio de una interfaz serial.
XRAM (Xmodem RAM) . -	Memoria RAM de 128 bytes definida en el sistema operativo del microcontrolador.
ZRAM . -	Memoria RAM de 512 bytes definida en el sistema operativo del microcontrolador.

CAPÍTULO 1

INTRODUCCIÓN

C O N T E N I D O

-
- 1.1 Antecedentes**
 - 1.2 Importancia**
 - 1.3 Alcance**
 - 1.4 Objetivos**
-

1.1 ANTECEDENTES

Desde la creación de los primeros sistemas de posicionamiento global por satélite (militar y civil), las aplicaciones sobre esta tecnología se han multiplicado en distintas áreas como son: navegación, transporte, geodésica, hidrográfica, agrícola, topográfica aeronáutica, náutica, entre otras. El desarrollo de la microelectrónica ha permitido la existencia en el mercado de unidades receptoras GPS de aplicación genérica, que permiten la implementación de sistemas propietarios basados en GPS.

En los últimos años, dispositivos portátiles de navegación y posicionamiento global, han aparecido en el mercado con un sin número de aplicaciones. En el Ecuador, estos dispositivos son importados desde otros países, los cuales incorporan en su memoria interna mapas de nuestras regiones, para ser utilizados en nuestro país. Estos mapas digitales son elaborados con software provisto por las marcas de los dispositivos.

En el Ecuador se han elaborado proyectos piloto de diseño de mapas digitales; como por ejemplo en la exploración de suelos en la ciudad de Guayaquil elaborado por la Escuela Politécnica del Litoral, donde se construyen mapas a partir de información geológica disponible. En el campo militar, el Instituto Oceanográfico de la Armada del Ecuador (INOCAR) realiza mapas digitales de inundación por tsunamis para ciertas zonas del país. Estos proyectos son realizados con software de cartografía digital como ArcGis y ArcView, con sistemas de archivos propietarios.

Hasta la fecha de realización del planteamiento de este proyecto, no existe en desarrollo sistemas de archivos de mapas digitales creados en nuestro país para aplicaciones en dispositivos de uso masivo como microcontroladores, que puedan ser almacenados y compartidos en una memoria tipo flash SD.

1.2 IMPORTANCIA

El presente proyecto es importante porque contribuye con la investigación y desarrollo en el campo de la electrónica, al plantearse la creación de un nuevo formato de archivo de mapa digital para navegación GPS, que posea las características necesarias para ser manejado por un microcontrolador, con lo que se abren nuevas expectativas para aplicaciones de geo-localización, con dispositivos ampliamente usados en la actualidad como son los microcontroladores.

1.3 ALCANCE

Se desarrolla un nuevo formato de archivo de mapa digital geo-referenciado, con la característica de que éste pueda ser leído e interpretado por un sistema basado en microcontrolador.

Se implementa un prototipo electrónico para navegación y localización global, en una plataforma microcontrolada, que sea capaz de interpretar los archivos digitales geo-referenciados que se encuentren en una memoria externa flash SD de una zona geográfica de prueba y relacionarlos con la posición actual mediante un receptor GPS. El sistema procesa la información, genera el mapa digital de calles y avenidas de la zona de prueba, las cuales se muestran en pantalla gráfica LCD de matriz de puntos monocromática.

1.4 OBJETIVOS

1.4.1 General

Diseñar e implementar un prototipo electrónico para navegación GPS basado en el desarrollo de un formato de archivo de mapa digital geo-

referenciado capaz de ser manejado por un microcontrolador y leído desde una memoria flash SD.

1.4.2 Específicos

- Realizar un estudio general del funcionamiento de los sistemas de navegación global por satélite (GPS en particular), historia, evolución y aplicaciones.
- Realizar un análisis de los protocolos de comunicación del receptor GPS, y de los datos generados por el mismo.
- Analizar el protocolo de comunicación aplicado a las memorias tipo flash SD, así como la implementación de un sistema de lectura de datos desde un microcontrolador.
- Desarrollar un nuevo formato de mapa digital capaz de ser manejado e interpretado por un sistema basado en microcontrolador.
- Crear archivos de mapas digitales de una zona de prueba y almacenarlos en una memoria flash SD.
- Desarrollar un sistema de generación de gráficos que permita la interpretación y manipulación de la información de mapas digitales de navegación GPS.
- Diseñar e implementar el hardware del dispositivo de navegación GPS basado en microcontrolador.
- Documentar el proyecto.

CAPÍTULO 2

MARCO TEÓRICO

C O N T E N I D O

- 2.1 Sistema de navegación global por satélite GPS**
 - 2.2 Receptores GPS de aplicación genérica**
 - 2.3 Mapas digitales aplicados a la navegación GPS**
 - 2.4 Estructura y funcionamiento de las memorias SD (*Secure Digital*)**
 - 2.5 Pantalla LCD de gráficos (GLCD)**
 - 2.6 Microcontroladores: recursos de comunicación y herramientas de desarrollo**
-

2.1 SISTEMA DE NAVEGACIÓN GLOBAL POR SATÉLITE GPS: HISTORIA, EVOLUCIÓN Y TEORÍA DE FUNCIONAMIENTO.

2.1.1 Historia de la navegación por satélite

La navegación con ayuda de aparatos e instrumentos ocurrió tempranamente en la historia del hombre con el descubrimiento de la brújula, que era utilizada en los viajes hacia lejanas tierras y también en guerras en los tiempos antes de la historia escrita. En la época moderna el impulso principal para desarrollar técnicas de navegación asistida por instrumentos sigue siendo principalmente el bélico, por esta razón el uso de la navegación por satélite comenzó siendo militar.

En 1957 la Ex URSS¹ lanzó al espacio el satélite *Sputnik I*, cuya posición era monitoreada por medio de la observación del efecto *doppler* de la señal transmitida desde el satélite. Fue entonces que los científicos se dieron cuenta que por medio de este mismo efecto, a partir de la posición de la órbita del satélite (que puede ser predicha con cálculos), se podía encontrar la posición de un receptor en tierra. Debido a este descubrimiento, los gobiernos de EUA² y Ex URSS rápidamente aplicaron esta tecnología, para proveer a los sistemas de navegación de sus flotas de observaciones de posiciones actualizadas y precisas.

A Principios de la II Guerra Mundial, se desarrollaron los primeros proyectos de posicionamiento hiperbólico por radio transmisores de baja frecuencia (en el orden de 90 a 110Khz). Por ejemplo *DECCA* del Reino Unido, *OMEGA* y *LORAN* de los Estados Unidos, *ALPHA* y *CHAYKA* de la Ex URSS.

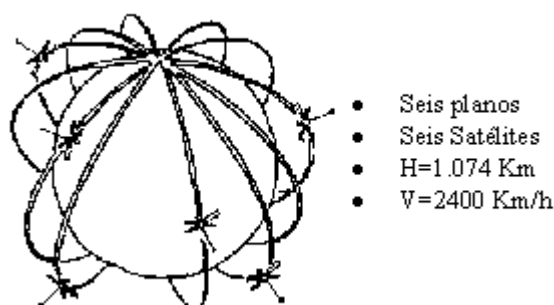
Los departamentos de defensa, transporte y la agencia espacial norteamericanas, desarrollan tres experimentos de navegación por satélite: *TRANSIT*, *TIMATION* y el proyecto Fuerza Aérea 621B, utilizando el efecto *doppler*, relojes atómicos y secuencias de pseudo-ruido.

(1) URSS: Unión de Repúblicas Socialistas Soviéticas

(2) EUA: Estados Unidos de América

En 1964 *TRANSIT* quedó operativo y estaba constituido por una constelación de seis satélites en órbita polar baja, a una altura de 1074 Km. *TRANSIT* trabajaba con dos señales en dos frecuencias, para evitar los errores debidos a la perturbación ionosférica. El cálculo de la posición se basaba en la medida continua de la desviación de frecuencia *doppler* de la señal recibida y su posterior comparación con tablas y gráficos. El error de *TRANSIT* estaba en torno a los 250 m. Su gran aplicación fue la navegación de submarinos y de barcos.

Figura 2.1 Constelación *TRANSIT*¹



La configuración conseguía una cobertura mundial pero no constante. La posibilidad de posicionarse era intermitente, pudiéndose acceder a los satélites cada 1.5 horas. El cálculo de la posición requería estar siguiendo al satélite durante quince minutos continuamente. La Ex URSS tenía un sistema igual que el *TRANSIT*, de nombre *TSICADA*. Los dos sistemas tenían problemas y con el comienzo de la guerra fría había que dar un gran salto en costes de investigación y desarrollo para mejorar la precisión y el tiempo de adquisición de los datos.

Estados Unidos da un gran paso que definitivamente deja a la Ex URSS atrás. Concibe entonces un sistema formado por 24 satélites en órbita media, que daría cobertura global y continua, utilizando todas las mejoras tecnológicas logradas con los experimentos militares anteriores dando paso a lo que se conoció como *Programa de Tecnología de Navegación*, posteriormente renombrado como *NAVSTAR GPS*².

(1) Fuente: www.alsitel.com/tecnico/gps/historia.htm

(2) GPS: Global Navigation System

El primer satélite se lanzó en 1978, y se planificó tener la constelación completa ocho años después. Unido a varios retrasos, el desastre del *Challenger* paró el proyecto durante tres años. En diciembre de 1983 se declaró la fase operativa inicial del sistema GPS. El objetivo del sistema GPS era ofrecer a las fuerzas de los EUA la posibilidad de posicionarse (disponer de la posición geográfica) de forma autónoma o individual, de vehículos o de armamento.

El uso exclusivamente militar del GPS cambiaría a partir de 1984 cuando ocurrió una tragedia. El 1 de septiembre de 1983, el vuelo KAL007 de *Korean Airlines* se salió de su ruta, penetrando en el espacio aéreo de la Ex URSS, tras lo cual fue abatido por un avión de combate soviético SU-15. Los 269 pasajeros y la tripulación murieron. Dos semanas más tarde, el presidente estadounidense Reagan propuso que el GPS se pusiera a disposición de los civiles para evitar que errores de navegación como ese provocasen catástrofes de tal calibre.

Entre 1978 y 1985 se desarrollaron y lanzaron once satélites prototipo experimentales NAVSTAR, a los que siguieron otras generaciones de satélites, hasta completar la constelación actual, a la que se declaró con “capacidad operacional inicial” en diciembre de 1993 y con “capacidad operacional total” en abril de 1995.

Debido al carácter militar del sistema GPS, el Departamento de Defensa de los EUA se reservaba la posibilidad de incluir un cierto grado de error aleatorio, que podía variar de los 15 a los 100 m. La llamada disponibilidad selectiva (SA)¹ fue eliminada el 2 de mayo de 2000. Aunque actualmente no aplique tal error inducido, la precisión intrínseca del sistema GPS depende del número de satélites visibles en un momento y posición determinados.

Existen actualmente varios sistemas de posicionamiento global similares al GPS, por ejemplo la contraparte Rusa *GLONASS*² ó Sistema de Navegación Global por Satélite Orbitante, que fue puesto en operación en Diciembre de 1995. Utiliza también una flota de 24 satélites y es ligeramente mas preciso que el GPS, pero en los años posteriores con la crisis Rusa, la constelación fue

(1) SA: Selective Availability

(2) GLONASS: GLObalnaya NAVigatsionnaya Sputnikovaya Sistema

actualizada ni reemplazada, con lo que gradualmente ha ido perdiendo efectividad. Se espera un resurgimiento total del sistema para el año 2012.

GALILEO es el sistema de posicionamiento por satélite de la Unión Europea, y fue desarrollado para no depender del sistema de Estados Unidos (GPS). GALILEO proveerá de mucha mayor exactitud que el GPS y se completará en el 2010 ó 2011. El primer satélite fue lanzado en 1995 y su flota constará eventualmente de 30 satélites orbitando la tierra a 23.600 Km de altura.

COMPASS (Bei Dou) es el proyecto GPS Chino, y consiste de 30 satélites a 21550Km de altura. Se utilizan 4 portadoras para transmitir las señales. El proyecto estará operativo en el año 2012.

2.1.2 Evolución del sistema GPS

GPS DIFERENCIAL (DGPS)¹

El sistema DGPS o GPS diferencial es un sistema que proporciona a los receptores de GPS correcciones de los datos recibidos de los satélites GPS, con el fin de proporcionar una mayor precisión en la posición calculada. Se concibió fundamentalmente debido a la introducción de la disponibilidad selectiva (SA).

Una estación DGPS consiste en un receptor GPS fijo en tierra (referencia) que conoce exactamente su posición basándose en otras técnicas, recibe la posición dada por el sistema GPS, y puede calcular los errores producidos por el sistema GPS, comparándola con la suya, conocida de antemano. Este receptor transmite la corrección de errores a los receptores próximos a él, y así estos pueden, a su vez, corregir también los errores producidos por el sistema dentro del área de cobertura de transmisión de señales del equipo GPS de referencia.

(1) DGPS: Differential Global Positioning System

Existen varias formas de obtener las correcciones DGPS. Las más usadas son:

1. Recibidas por radio, a través de algún canal preparado para ello, como el RDS en una emisora de FM.
2. Descargadas de Internet, o con una conexión inalámbrica.
3. Proporcionadas por algún sistema de satélites diseñado para tal efecto.

Con el DGPS se pueden corregir en parte los errores debidos a:

- Disponibilidad selectiva (eliminada a partir del año 2000)
- Propagación por la ionosfera - troposfera
- Errores en la posición del satélite
- Errores producidos por problemas en el reloj del satélite

Para que las correcciones DGPS sean válidas, el receptor tiene que estar relativamente cerca de alguna estación DGPS; generalmente, a menos de 1.000 km. La precisión lograda puede ser de unos dos metros en latitud y longitud, y unos tres metros en altitud. Este sistema no es aplicable en nuestro medio puesto que no existe una estación DGPS cercana.

SISTEMAS DE AUMENTACIÓN BASADOS EN SATÉLITES

Los sistemas de aumentación basados en satélites están ideados para ser un complemento a los sistemas ya existentes de navegación por satélite, proporcionando datos de corrección para la posición de los receptores en tierra y así obtener una mayor exactitud en las lecturas (en el orden de 2 ó 3 metros).

Estos sistemas de aumentación se constan de flotas de estaciones transmisoras y receptoras de datos, llamadas estaciones de referencia, que funcionan en conjunto con satélites geoestacionarios complementarios pero distintos de los de posicionamiento global.

Dependiendo del sistema las estaciones se ubican entre 800 y 1000Km de distancia entre ellas y su implementación y operación están a cargo de cada grupo de países, país o región del planeta. Todos los sistemas de aumentación de uso civil existentes son compatibles entre sí y son WAAS, EGNOS, MSAS, GAGAN.

WAAS (Wide Area Augmentation System)

Desarrollado por los Estados Unidos, y consta de 24 estaciones de vigilancia o referencia y 3 satélites geoestacionarios. Éstas estaciones de vigilancia se encuentran a lo largo de los Estados Unidos, Alaska, Hawái y Puerto Rico.

Las correcciones WAAS (mensajes de 250 bits) recibidas son aplicados a los resultados de los cálculos de pseudo-distancia para luego determinar la posición.

EGNOS (European Geostationary Navigation Overlay Service)

Es un Sistema desarrollado por la Agencia Espacial Europea (ESA), la Comisión Europea (institución de la Unión Europea) y Eurocontrol, y consiste en una red de 3 satélites geoestacionarios y 34 estaciones de referencia.

MSAS (Multi-functional Satellite Augmentation System)

Es un Sistema de Aumentación Basado en Satélites desarrollado en el Japón el cual soporta DGPS, e incluye dos modos de comunicación de voz y datos. Esta capacidad de comunicación será usada para brindar vigilancia automática dependiente.

GAGAN (GPS and GEO Augmented Navigation)

Es un Sistema de Aumentación desarrollado por la Agencia India del Espacio (ISRO) y se ha diseñado para permitir relacionarse con WAAS, EGNOS y MSAS y provee una precisión de 3 metros.

GPS III

El GPS III es la versión nueva del GPS que enviará una señal más potente hacia la tierra de tal manera que en países con una alta latitud puedan utilizar este servicio que anteriormente no era posible. La mayor ventaja es que puede operar en coordinación con el sistema GALILEO para brindar una mayor exactitud.

Entre las ventajas principales frente a su predecesor tenemos:

- Incorporación de una nueva señal en L2 para uso civil
- Adición de una tercera señal civil (L5): 1176,45 MHz
- Protección y disponibilidad de una de las dos nuevas señales para servicios de Seguridad Para la Vida (SOL).
- Mejora en la estructura de señales.
- Incremento en la potencia de señal (L5 tendrá un nivel de potencia de -154 dBW en la superficie de la tierra, alrededor de 10dB mayor).
- Mejora en la precisión (1 – 5 m).
- Aumento en el número de estaciones monitorizadas: 12 (el doble)
- Permitir mejor interoperabilidad con la frecuencia L1 de Galileo

El programa GPS III persigue el objetivo de garantizar que el GPS satisfará requisitos militares y civiles previstos para los próximos 30 años. Este programa se está desarrollando para utilizar un enfoque en 3 etapas (una de las etapas de transición es el GPS II); muy flexible, permite cambios futuros y reduce riesgos. El desarrollo de satélites GPS II comenzó en 2005, y el primero de ellos estará disponible para su lanzamiento en 2012, con el objetivo de lograr la transición completa de GPS III en 2017.

2.2.3 Funcionamiento del sistema GPS

Figura 2.2 Satélite Navstar-2F¹



El sistema GPS está formado por tres segmentos principales: El segmento espacial, el segmento de control, y el segmento de usuario.

El segmento espacial tiene por objetivo que a partir de las instrucciones que reciben del segmento de control: proporcionar una referencia de tiempo atómico, generar secuencias de códigos, almacenar y reenviar el mensaje de navegación. El segmento espacial está formado por una constelación de 24 satélites operativos, uniformemente dispersados alrededor de la Tierra en 6 órbitas a una altitud media de 20.200 Km. Los 6 planos orbitales están inclinados 55° sobre el plano del ecuador y la separación entre las órbitas es de 60° . Las órbitas son casi circulares, tienen una excentricidad menor a 0,02, con un semieje mayor de unos 26.000 Km. y un periodo de 12 horas sidéreas (11h 58min 2seg).

El segmento de control está formado por estaciones de tierra que se encargan del control y mantenimiento del estado y la configuración de la constelación de satélites. También es el encargado de predecir el comportamiento de los relojes de los satélites, mantener la escala de tiempo GPS, y actualizar periódicamente el mensaje de navegación de cada uno de los satélites. El segmento está formado por 3 componentes físicos diferentes:

(1) Fuente: <http://commons.wikimedia.org/wiki/File:Navstar-2F.jpg>

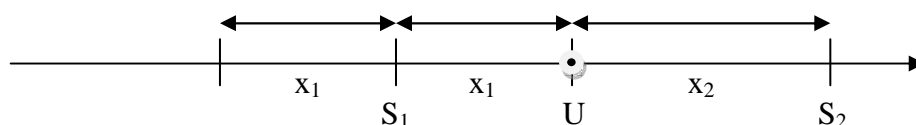
- *Master Control Station* (MCS) ó Estación de Control Principal.
- *Monitor Stations* ó Estaciones de Monitoreo.
- *Ground Control Stations* ó Estaciones de Control de Tierra.

El segmento usuario lo componen los instrumentos de recepción que utiliza el usuario para realizar la acción de posicionamiento y navegación. Los elementos que forman el equipo de observación lo constituyen: la antena, el receptor, y el controlador que contiene el software para realizar el cálculo de los parámetros GPS.

CONCEPTO BÁSICO DE POSICIONAMIENTO

La posición de un punto en el espacio puede ser encontrada midiendo la distancia desde este punto hacia algunas referencias o posiciones conocidas en el espacio. Por ejemplo en la siguiente figura se muestra el caso unidimensional para la determinación de la posición de usuario U a partir de posiciones conocidas S_1 y S_2 .

Figura 2.3 Posición de usuario en el caso unidimensional

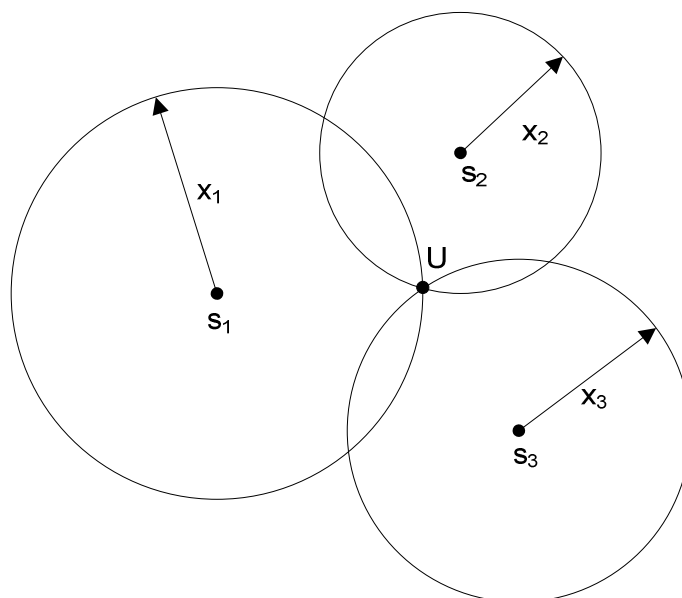


En este caso la posición del satélite S_1 y su distancia X_1 son conocidas. Sin embargo existen dos posibles soluciones para que la posición de usuario sea determinada. Es necesario entonces la existencia de un segundo satélite con posición y distancia conocidas S_2 y X_2 , entonces en la intersección de estas dos distancias se podrá saber con exactitud la posición de usuario U .

En el caso bidimensional, con el fin de determinar un punto en el plano, tres satélites y tres distancias son requeridas. El trazado desde un punto con distancia constante hacia su alrededor resulta una circunferencia con radio

igual a la distancia del satélite. La intersección de dos circunferencias resulta en dos puntos dando dos posibles soluciones. Es necesario entonces una tercera circunferencia para hallar la posición exacta como se ilustra en el siguiente gráfico:

Figura 2.4 Posición de usuario en el caso bidimensional



En el caso tridimensional la distancia constante desde un punto hacia el espacio es una esfera. La intersección de dos esferas forman un círculo, con tres esferas tenemos dos puntos posibles. La posición podría determinarse solamente con estos tres satélites debido a que una de las soluciones corresponde a un punto en el espacio y la otra en la superficie de la tierra.

Sin embargo un cuarto satélite es necesario para la determinación exacta de la posición de usuario, debido al error de sincronización entre los relojes del GPS y de los satélites existentes; además de determinar la altura del usuario respecto al nivel del mar. Los receptores GPS utilizan relojes convencionales de cuarzo para la sincronización, puesto que si incorporaran relojes atómicos como los satélites, la navegación GPS no sería asequible por el costo que implica.

DISTANCIA DE USUARIO Y TRANSMISIÓN DE CÓDIGO

El sistema GPS se encuentra disponible en dos tipos de servicios, el primero conocido por sus iniciales SPS (*Standard Positioning Service*) ó Servicio de Posicionamiento Estándar, y el segundo por la sigla PPS (*Precise Positioning Service*) ó Servicio de Posicionamiento Preciso.

Para que estos dos servicios coexistan en el mismo sistema se transmiten dos señales en dos frecuencias de portadora distintas: L1 y L2 en la banda L (1 a 2 GHz). El servicio PPS de uso militar utiliza el código P (de precisión) y provee de máxima exactitud en la banda L2. El servicio SPS de uso civil con código C/A (clear/acquisition) que provee de menor exactitud dentro de la banda L1.

Estos dos componentes de frecuencia en los links L1 Y L2, localizados a 1575.42 MHz y 1227.6 MHz respectivamente, son coherentes con una señal de reloj generada de 10.23 MHz y se observa de la siguiente manera:

$$L1 = 1575.42 \text{ MHz} = 154 \times 10.23 \text{ MHz (civil)}$$

$$L2 = 1227.6 \text{ MHz} = 120 \times 10.23 \text{ MHz (militar)}$$

Estas frecuencias son generadas a partir de un reloj atómico muy preciso en los satélites. Para tomar en consideración el efecto relativista debido a las velocidades en cuestión, dicha frecuencia es ligeramente menor (aproximadamente 10.229999995433MHz). Los receptores GPS reciben esta señal pero con un cambio debido al efecto *doppler* que se produce por el movimiento del satélite y del receptor en tierra. Las desviaciones de frecuencia son de alrededor de +/- 5Khz y producirán variaciones en la señal recibida.

En términos muy generales para establecer la distancia de un satélite al usuario, el receptor genera la misma señal que envía el satélite con un mismo código conocido, retardando la señal hasta que coincida con la recibida y de esta manera se mide el retraso de la señal en tiempo, que multiplicado por la velocidad de la luz se obtiene una pseudodistancia.

Se denomina pseudodistancia debido a que no es una distancia verdadera, sino que está afectada por errores de distinta naturaleza como: precisión del reloj del receptor, cambio de medio de la señal electromagnética por las distintas capas de la atmósfera, multipath, etc.

Para que coexistan varias señales de satélite en una misma frecuencia y en un mismo lugar, y para proteger la integridad de los datos transmitidos, la aplicación de un código es necesaria. Por esto se han ideado los códigos C/A y P en la transmisión de datos de satélite GPS.

C/A (Coarse/Acquisition code)

También llamado código estándar, es un código de pseudo-ruido randómico o PRN (pseudo-random noise), con el cual se modulan los datos de GPS para uso civil en la banda L1 con un *chip rate* de 1023MHz, es decir la tasa de bits es de 1Mbps, y se repite cada milisegundo.

P (Presicion Code)

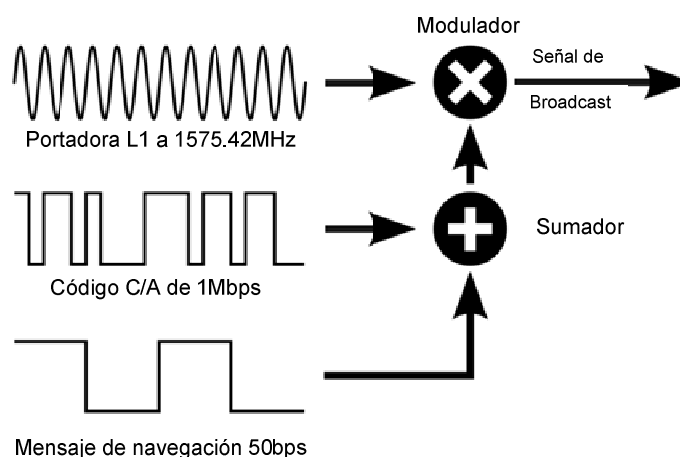
El código P o de precisión es un código de pseudo-ruido randómico o PRN (pseudo-random noise) de longitud 6.1871×10^{12} bits, transmitidos a 10Mbps, que se repiten una vez a la semana. La extrema longitud del código P incrementa la correlación por lo tanto la precisión de la distancia. Debido a la complejidad de sincronización del código por parte del receptor, es necesario sincronizarse primero con el código C/A (civil) para adquirir la fecha y hora exactas, y luego sincronizar el código P (militar).

Cada satélite transmite una porción del código maestro completo que tiene una longitud de 2.35×10^{14} bits. Para evitar una posible interferencia de la señal militar (*satellite spoofing*), el código P se modula con código W que es una secuencia especial de encriptación para generar el código Y, llamado también código P(Y). La generación de este código anti-interferencia (*anti-spoofing*) es mantenido en secreto por parte de la armada y el departamento de defensa de los Estados Unidos.

Mensaje de navegación

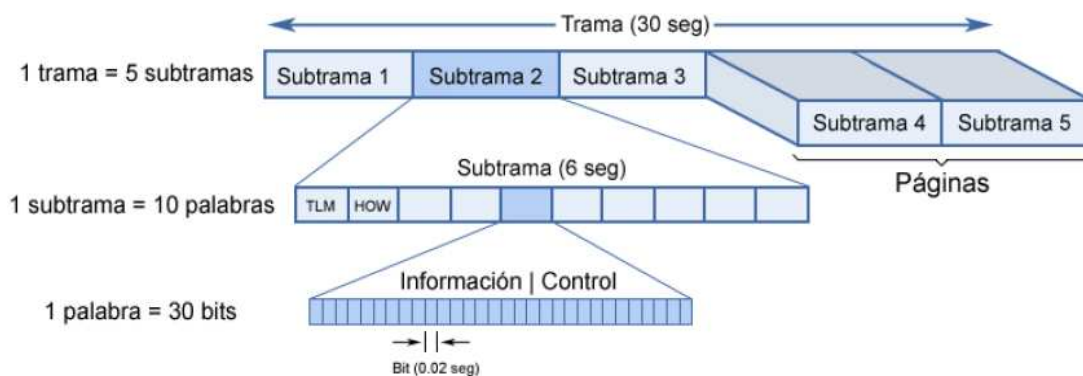
El método de acceso al medio es de naturaleza *Spread Spectrum* (Espectro Disperso), es decir que el acceso es CDMA y los datos GPS que llevan estos códigos corresponden al mensaje de navegación que se modula sobre las portadoras L1 y L2 a 50bps.

Figura 2.5 Ejemplo de modulación sobre la portadora L1



El mensaje tiene una duración de 12 minutos y 30 segundos y consta de 25 tramas de 1500 bits cada uno, con lo que una trama se transmite en 30 segundos. Cada trama está dividida en 5 subtramas de 300 bits cada una, que a su vez se dividen en 10 palabras de 30 bits de longitud (siguiente figura).

Figura 2.6 Formato básico del mensaje de navegación GPS ¹



(1) Fuente: <http://upcommons.upc.edu/pfc/bitstream/2099.1/3827/1/53880-1.pdf>

Cada subtrama empieza con la palabra telemetría (TLM o TeLeMetry), que ofrece 14 bits de información sobre el estado del satélite, siendo el resto bits de sincronización y de paridad. La segunda palabra, llamada de transferencia (HOW o *Hand Over Word*), contiene el tiempo de la semana (TOW o time-of-week), un contador de 17 bits, que permite determinar el tiempo transcurrido desde que se inició el código P. Esto, utilizado junto al código C/A, permite una rápida sincronización con el código P.

El contenido de cada una de las subtramas es:

- *Subtrama 1:* contiene información relativa a las correcciones del reloj del satélite (los coeficientes polinómicos para convertir el tiempo de a bordo en tiempo GPS), la salud del satélite, la antigüedad de la información y otras. Se repite cada 30 segundos, es decir, se renueva a cada nueva trama, y es una información que el segmento de control transmite a cada satélite.
 - *Subtrama 2 y 3:* contiene las efemérides del satélite, que permiten un cálculo preciso de su posición y velocidad, así como la edad de dichos parámetros.
 - *Subtrama 4:* Se encuentran los parámetros de modelo ionosférico, información del tiempo UTC, parte del almanaque e indicaciones de si está activado la Disponibilidad Selectiva. Mucha información que incluye está reservada para uso militar.
 - *Subtrama 5:* contiene los datos del almanaque y el estado de la constelación. Con ello se permite una rápida identificación de los satélites de los que procede la señal. Se precisan 25 tramas para completar el almanaque.
-

2.2 RECEPTORES GPS DE APLICACIÓN GENÉRICA Y PROTOCOLOS DE COMUNICACIÓN UTILIZADOS

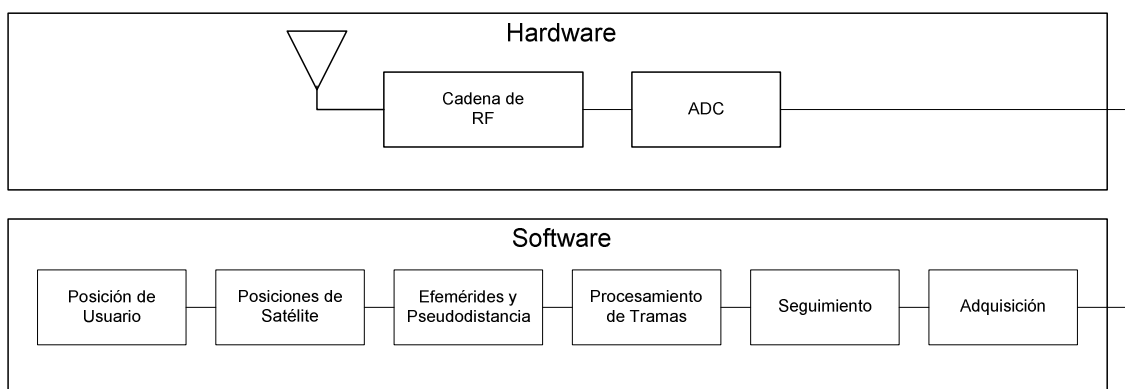
2.2.1 Introducción

Gracias al desarrollo de la microelectrónica y miniaturización ahora se dispone de receptores de GPS muy pequeños. Se incorporan muchas veces en teléfonos celulares y dispositivos portátiles.

Los receptores GPS poseen una unidad receptora con una antena, una cadena de RF (Radio Frecuencia) y un microprocesador, el cual efectúa el control y procesamiento de datos. El procesador se encarga de generar las secuencias pseudoaleatorias, realizar los cálculos de pseudo-distancias, altura, velocidad, etc.

El receptor de GPS se puede dividir en dos partes principales: Hardware y Software. A continuación se observa un gráfico en bloques del dispositivo:

Figura 2.7 Diagrama de bloques del receptor GPS



Para realizar los cálculos de posicionamiento, el receptor GPS debe tener la información necesaria acerca del estado de los miembros de la constelación y la posición de cada uno con el fin de predecir la órbita de los satélites NAVSTAR. Esta información se denomina “efemerides” (*ephemeris*) y “almanaque”, y el receptor la obtiene de los mismos satélites.

Debido a la actualización de datos de almanaque y efemérides, los receptores GPS tardan un tiempo en inicializarse, dependiendo si la información se encuentra ó no en el receptor. Por esta razón los fabricantes indican el tiempo de inicialización con tres modos básicamente: *Cold Start*, *Warm Start*, y *Hot Start*, siendo *Cold Start* ó el tipo de arranque mas largo y *Hot Start* el de menor duración.

EFEMÉRIDES (*Ephemeris*)

Las efemérides es la información transmitida por los satélites que indica la posición exacta de los mismos. El GPS usa los datos de efemérides de varios satélites al mismo tiempo para realizar cálculos y correcciones muy complejas que le permiten encontrar cuál es su posición en la Tierra.

Cada satélite solo envía sus propios efemérides. Esta información cambia frecuentemente, siendo actualizada por las estaciones de seguimiento de la Tierra. Los parámetros orbitales de los satélites se van actualizando a medida que su movimiento se ve alterado por la atracción del Sol y la Luna, la diferencia de gravedad entre distintas zonas de la corteza terrestre, viento solar, etc.

La validez de los datos contenidos en las efemérides depende de cada satélite en particular, pero pueden llegar a ser útiles hasta 4 horas. Las efemérides son bloques de información digital que el receptor GPS tarda 30 segundos en recibir, y son enviados de forma continua (es decir, se reenvían de nuevo cada 30 segundos). De esta forma un receptor GPS siempre tendrá muchas oportunidades para recibir esta información.

ALMANAQUE (*Almanac*)

El almanaque está constituido por un conjunto de valores o parámetros que le permiten al receptor GPS predecir de manera aproximada la órbita y la posición de todos los satélites, informando sobre ellos mismos y el resto de

satélites miembros del sistema, su nivel de salud, etc. Esta información suele variar con poca frecuencia en el año. Cada satélite envía de forma individual los datos correspondientes al almanaque de todos los demás satélites. Los datos de este almanaque son útiles durante varios meses.

WGS84

WGS84 son las siglas de *World Geodetic System 1984* ó sistema de coordenadas mundiales geodésicas, que data de 1984. Es un sistema de referencia terrestre con punto referencial geocéntrico fijo con la Tierra; muy utilizado en cartografía, geodesia y navegación. Las coordenadas geodésicas (WGS84) son diferentes de las coordenadas planas porque toman en cuenta la forma de la tierra (geoide) que es un esferoide es decir una esfera con los polos achatados debido al movimiento rotacional, y además se consideran variaciones determinadas por gravimetría, las cuales indican que el campo gravitacional no es uniforme en todo el planeta, sino que está afectado por mareas, atracción de la luna, el sol y los planetas, estaciones del año, etc.

2.2.2 El receptor GPS GlobalSat BR-355

El receptor GPS utilizado en el presente proyecto es el GlobalSat BR-355 de comunicación serial. Se alimenta de 5V y posee circuitería “SiRF Star II”. El procesamiento paralelo de 12 canales permite obtener datos de varios satélites al mismo tiempo. Utiliza el protocolo NMEA0183 para la comunicación de datos con sentencias como: GGA, GSA, GSV, RMC, GLL y VTG.

Figura. 2.8 Receptor GPS GlobalSat BR-355 con conector PS/2 ¹



(1) Fuente: www.navpal.co.uk

La ubicación de un led indicador de estado permite visualizar cuando el receptor está enganchado o fijo (led parpadeante), y cuando el receptor está buscando señal o no se encuentra enganchado (led encendido).

La interface eléctrica se realiza por medio de un conector PS/2 macho y comunicación serial RS-232. La distribución de pines en el conector es una variante del estándar de conexión de mouse o teclado PS/2. La especificación y comparación de pines de los dos sistemas se detalla en el capítulo 5.

Especificaciones del sistema (receptor)	
Frecuencia	L1, 1575.42 MHz
Código C/A	1.023 MHz chip rate
Canales	12
Sensibilidad	-170 dBW
Velocidad	0.1m/seg 95% (DS apagado),
Tiempo	1 micro-segundo sincronizado con tiempo GPS
WAAS	disponible
Tipo de datos	WGS-84
Rata de adquisición	Hot start 8 seg., promedio (con efemérides y almanaque válidos) Warm start 38 seg., promedio (con almanaque pero no efemérides) Cold start 45 seg., promedio (sin almanaque ni efemérides)
Readquisición	0.1 sec. promedio
Protocolo	Por Defecto: NMEA 0183 (Secundario: SiRF binario)
Interfaz de comunicación	Por Defecto : 4800,n,8,1 para NMEA
Límite de aceleración	Hasta 4g
Límite de altitud	18000 metros (60000 pies) max.
Límite de velocidad	515 metros/seg. (1000 nudos) max.
Voltaje	5V \pm 5%
Corriente	80mA típico (modo continuo)
Dimensiones	59mm x 47mm x 21mm

Tabla 2.1 Resumen de especificaciones del receptor GPS GlobalSat BR-355

2.2.3 El protocolo NMEA-0183

La Asociación Nacional de Electrónica para la Marina NMEA (*National Marine Electronics Association*), es una asociación sin fines de lucro de fabricantes, distribuidores, instituciones educativas y otros interesados en equipos periféricos marinos. La definición estándar de un NMEA 0183 es una interfaz eléctrica y un protocolo de datos para la comunicación entre instrumentos marinos. NMEA se establece como un grupo de trabajo para el desarrollo de nuevos estándares de comunicación de datos para dispositivos marinos a bordo de las embarcaciones.

Este protocolo se lanza por primera vez en marzo de 1983, y su última versión fue publicada en el 2001.

El protocolo NMEA es usado para la comunicación entre dispositivos de uso marino para transmitir datos. La salida NMEA cumple con la recomendación EIA-422A, pero para la mayoría de los propósitos puede considerarlo RS-232 compatible.

Los equipos que generalmente utilizan un protocolo NMEA son:

- GPS
 - Brújula digital
 - Radares
 - Ecosonda, profundidad
 - Sensores de velocidad: magnéticos, *doppler* o mecánicos
 - Instrumentos meteorológicos
 - Transductores
 - Relojes atómicos
 - Sistemas de navegación integrados
 - Comunicaciones satelitales o de radio
-

INTERFÁZ ELÉCTRICA

Los equipos que utilizan NMEA son dispositivos unidireccionales o bidireccionales donde se emplea una interfaz asíncrona serial con los siguientes parámetros:

- Baud Rate: 4800
- Bits de datos: 8
- Bits de parada: 1 ó más
- Paridad: Ninguna
- Handshake: Ninguno

FORMATO DE SENTENCIA

Existen tres tipos de sentencias NMEA y son: Envío (*Talker Sentences*), Origen del Equipo (*Proprietary Sentences*) y Consulta (*Query Sentences*). Las sentencias de envío son de especial relevancia puesto que son las sentencias que maneja el receptor GPS y su formato general es como sigue:

$$\$ttsss,d1,d2,\dots<CR><LF>$$

Las primeras dos letras después de “\$” son el identificador de envío, los siguientes tres caracteres “sss” corresponden al identificador de sentencia seguidos de cierto número de datos separados por comas seguidos de un carácter “*” y una suma de verificación opcional correspondiente a la suma exclusiva lógica de todos los miembros de la sentencia excepto las comas, que luego es terminada con un <CR> y <LF> (*Carriage Return* ó Retorno de Carro y *Line Feed* ó Avance de Línea).

Los identificadores más comunes utilizados en GPS son: GGA, GLL, GSA, GSV, RMC, VTG.

ANÁLISIS DE LA TRAMA DE DATOS GPRMC

La trama de datos GPRMC con identificador RMC corresponde a la “Información mínima de navegación recomendada” y es una de las sentencias enviadas por el receptor GPS que tiene especial relevancia en este proyecto. A continuación se presenta el detalle de la sentencia que se utilizará para permitir el posicionamiento y la navegación y se explicarán cada uno de sus componentes:

RMC: Información Mínima De Navegación Recomendada

$$\begin{array}{cccccccccccc} & & 2 & & 4 & & 6 & & 9 & & 11 & \\ & & | & & | & & | & & | & & | & \\ \$GPRMC, & \underline{hhmmss.sss}, & E, & \underline{ggmm.ssss}, & L, & \underline{gggmm.ssss}, & L, & \underline{x.x}, & \underline{x.x}, & \underline{xxxx}, & \underline{x.x}, & a^* \underline{hh} \\ & 1 & & 3 & & 5 & & 7 & 8 & & 10 & 12 \end{array}$$

- 1) Hora (UTC)
- 2) Estado, V = Receptor en Alerta, A = Información Válida
- 3) Latitud
- 4) N o S
- 5) Longitud
- 6) E o W
- 7) Velocidad sobre fondo en Nudos (knots)
- 8) Track made good, en grados verdaderos
- 9) Fecha, día-mes-año (ddmmyy)
- 10) Variación Magnética, grados
- 11) E o W
- 12) Suma de Verificación (Checksum)

Hora UTC (hhmmss.sss)

Corresponde a la hora del tiempo universal coordinado o UTC, también conocido como tiempo civil, que es el tiempo de la zona horaria de referencia respecto a la cual se calculan todas las otras zonas del mundo. Es el sucesor del GMT (Greenwich Mean Time: tiempo promedio del Observatorio de Greenwich, en Londres). Los datos tienen una precisión de 1 milisegundo.

Estado (V ó A)

Se refiere al estado del receptor en cuanto a datos disponibles. La letra A significa que el receptor se encuentra en alerta, no se ha completado la

información para calcular la posición, aunque se existen disponibles la fecha y hora. Si el receptor se encuentra en V significa que todos los datos de posicionamiento y navegación se encuentran disponibles.

Latitud (N o S)

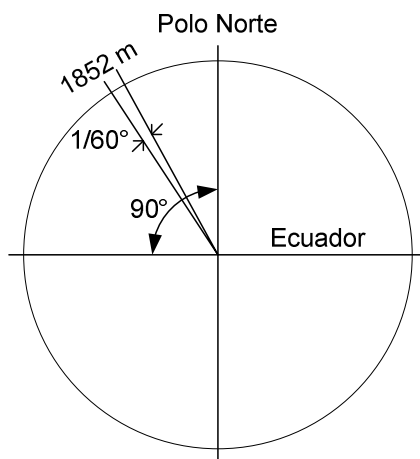
Corresponde a la latitud actual que puede variar entre 0 y 90 grados. El siguiente dato corresponde al hemisferio Norte o Sur.

Longitud (E o W)

Corresponde a la longitud actual que puede variar entre 0 y 180 grados. El siguiente dato corresponde al hemisferio Este u Oeste (E ó W).

Velocidad sobre fondo (speed over ground)

Figura 2.9 Gráfica de explicación de una milla náutica

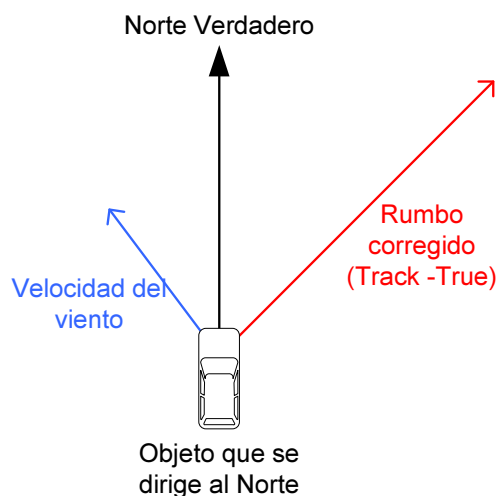


Es la velocidad del receptor respecto a la tierra fija que proporciona el GPS. En esta trama la velocidad está en Nudos (Knots - KN abreviado) que es una unidad de velocidad utilizada en meteorología, navegación marítima y aérea, equivalente a una milla náutica por hora. Un nudo equivale a 1852Km/h ó 0.514 m/s. Una milla náutica es una unidad de longitud y equivale a un minuto de arco de latitud paralelo a cualquier meridiano.

Track made good, en grados verdaderos

En aeronáutica “*Track made good*” es el vector de corrección de rumbo debido al viento que se expresa en grados verdaderos, y corresponde al vector de direccionamiento para llegar a cierto punto menos el vector de la velocidad del viento. En el presente proyecto no se hace uso de este parámetro.

Figura 2.10 Ejemplo de vector “Track made good ó Track - True”



Fecha, día-mes-año (ddmmyy)

Como su nombre lo indica corresponde a la fecha UTC expresado en día mes y año.

Variación magnética, grados (E o W)

La variación ó declinación magnética en un punto de la tierra es el ángulo comprendido entre el norte magnético local y el norte verdadero. En otras palabras, es la diferencia entre el norte geográfico y el indicado por una brújula (el denominado también norte magnético). La declinación puede ser hacia el este o al oeste. En el presente proyecto no se hace uso de este parámetro.

Suma de verificación (checksum)

Corresponde a la suma exclusiva lógica de todos los miembros de la sentencia exceptuando las comas, y sirve para identificar la existencia de un error en la sentencia generada por el GPS al compararla con la *checksum* calculada por un controlador. En el presente proyecto no se hace uso de este parámetro, sin embargo es de utilidad el carácter “*” que precede a la suma de verificación como se observará mas adelante.

2.3 MAPAS DIGITALES APLICADOS A LA NAVEGACIÓN GPS

2.3.1 Introducción

La tecnología de posicionamiento global (GPS), los sistemas de información geográfica (GIS), y el internet han revolucionado la cartografía en general. Los datos geográficos son recolectados con herramientas digitales que proveen de información exacta y variada para crear mapas digitales de todo tipo. Los mapas digitales pueden incluir distintos tipos de información dentro de una misma plataforma, como: carreteras, ríos, fotografías, rutas, censos, historia, entre otros.

2.3.2 Tipos de mapas digitales

Existen básicamente dos tipos de mapas digitales: mapas convencionales (tipo *raster*) y mapas vectoriales.

Los mapas convencionales o *raster* son mapas basados en fotografías, es decir son nada más que archivos fotográficos adquiridos por escáner o cámara y por su naturaleza no pueden alojar datos adicionales. Su procesamiento es lento, pues la interfaz gráfica interpreta la fotografía y la mueve pixel por pixel en la pantalla. Ejemplo: Archivos BMP, GIF, JPG, etc.

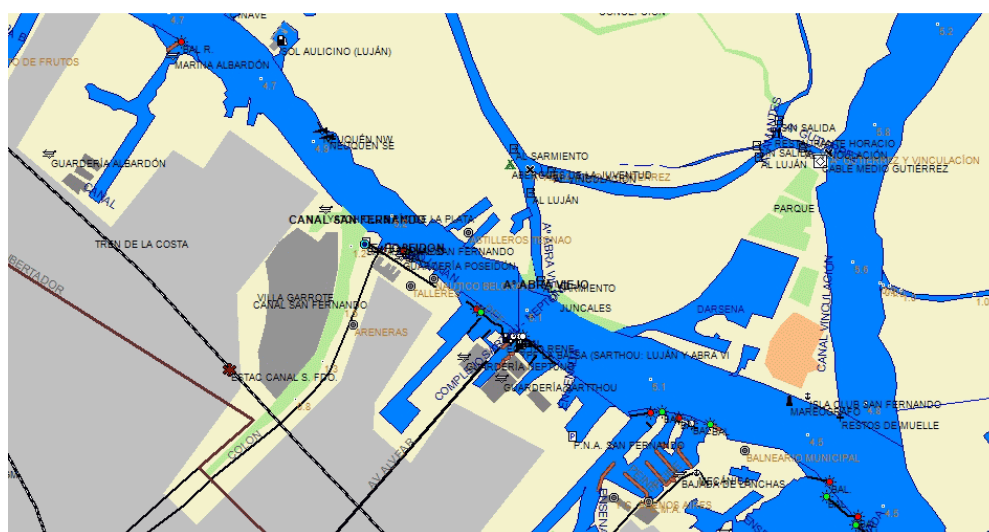
Los mapas vectoriales son archivos basados en puntos cuyas coordenadas se encuentran descritas en el archivo, y le dicen a la interfaz gráfica cómo interpretarlos, es decir un mapa vectorial es una imagen digital formada por objetos geométricos independientes (segmentos, polígonos, arcos, etc.), cada uno de ellos definido por distintos atributos matemáticos de forma, de posición, de color, etc. Por lo tanto este tipo de mapas son de menor peso y rápido manejo; incluso pueden llevar información adicional. Ejemplo: Archivos DWG (Autocad), AI (Adobe Illustrator), CDR (CorelDraw), etc.

Si se combina los dos tipos de mapas, surge un archivo híbrido que puede contener tanto la información vectorial y además se complementa con fotografías referenciadas. Ejemplo: Mapas de *Google Earth*, archivos PDF, etc.

2.3.3 Mapas digitales aplicados a la navegación GPS

Distintos fabricantes de software han creado una variedad de formatos de archivo para contener mapas vectoriales y *raster*, con información de navegación para ser utilizados con un GPS. Dicha información constituyen los mapas geo-referenciados, es decir cada objeto en el mapa guarda una relación con sus respectivas coordenadas geográficas.

Figura 2.11 Ejemplo de mapa digital vectorizado¹



(1) Fuente: <http://www.nauticaygps.com.ar/Archivos/digimaps/digimaps.php>

Debido a factores de marketing, el código interno y la estructura interna de los archivos de mapas digitales geo-referenciados no está disponible; es decir son archivos de código cerrado. En el presente proyecto se pretende crear desde cero un formato de archivo de mapa vectorial geo-referenciado para ser leído por el software de control del prototipo GNOM-4550. Esto se discute con detalle en el capítulo siguiente.

2.4 ESTRUCTURA Y FUNCIONAMIENTO DE LAS MEMORIAS SD

2.4.1 Introducción

Las memorias flash SD o *Secure Digital* son dispositivos de almacenamiento masivo de datos digitales no volátiles implementados sobre material semiconductor.

Las tarjetas SD se dividen en dos tipos principales: Estándar y HC (*High Capacity*) o alta capacidad. Existen tres tipos de memorias SD diferenciadas principalmente por su tamaño físico, y se detallan en la siguiente tabla:


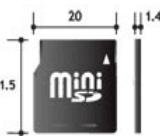
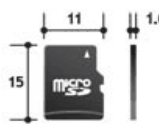
Tipo	SD	miniSD	microSD
Tamaño			
Area	768 mm ²	430 mm ²	165 mm ²
Volumen	1,613 mm ³	602 mm ³	165 mm ³
Espesor	2.1 mm	1.4 mm	1.0 mm
Peso	Aprox. 2g	Aprox. 1g	Aprox. 0.5g
Número de pines	9 pines	11 pines	8 pines
Sistema de archivos	FAT16/32	FAT16/32	FAT16/32
Voltaje de operación	2.7V - 3.6V	2.7V - 3.6V	2.7V - 3.6V
Protección contra escritura	SI	NO	NO
Protección copyright	SI	SI	SI
Compatibilidad	-	SI (con adaptador)	SI (con adaptador)
Capacidad (estándar)	Hasta 2 GB	Hasta 2 GB	Hasta 2 GB
Capacidad (HC)	4 - 32GB	4 - 32GB	4 - 32GB

Tabla 2.2 Características físicas de los tipos de memoria flash SD¹

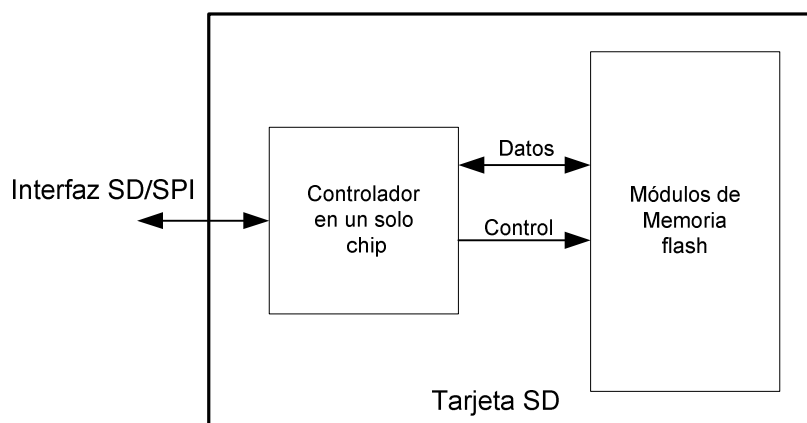
(1) Fuente: <http://www.sdcard.org/developers/tech/>

2.4.2 Estructura interna

La comunicación en tarjetas SD está basada en una interfaz de nueve pines (reloj, comandos, 4 líneas de datos, y 3 líneas de poder). La interfaz del host soporta la operación con tarjetas regulares MMC (Multi Media Card), es decir se ha guardado compatibilidad entre tarjetas donde mayormente solo son diferentes en el proceso de inicialización. Las especificaciones fueron originalmente definidas por Matsushita, Toshiba y La Corporación Sandisk. Actualmente son controladas por La Asociación Secure Digital (SDA).

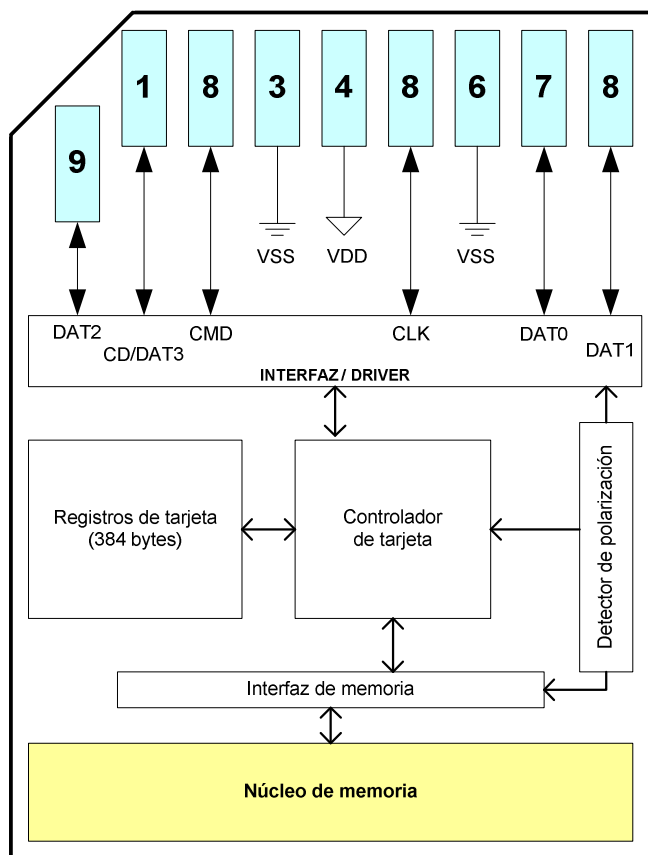
La estructura básica de la memoria SD se indica en el siguiente gráfico:

Figura 2.12 Estructura básica de la memoria SD



2.4.3 Arquitectura de la memoria

La arquitectura del sistema se describe por medio de bloques, donde el controlador principal de la tarjeta comanda todo el sistema. El bloque detector de polarización, se encarga de detectar y controlar el voltaje apropiado para el encendido de los demás bloques, así como reiniciar el sistema.

Figura 2.13 Arquitectura de la memoria SD

Los registros de la tarjeta trabajan conjuntamente con el controlador del dispositivo para alojar datos sobre la configuración del sistema, con 384 bytes internos. El bloque interfaz / driver permite acoplar las señales externas hacia el controlador así como amplificar adecuadamente las señales internas hacia el mundo exterior.

En la interfaz de memoria existe la circuitería apropiada para manejar los arreglos de bloques de memoria internos que están constituidos por el bloque “núcleo de memoria”.

2.4.4 Características principales

La familia de tarjetas SD soportan las siguientes características:

- Almacenamiento hasta 2GB en tarjetas estándar y hasta 32GB en HC

- Soporta protocolo SD (1bit y 4bit de transferencia)
- Soporta modalidad SPI
- Rango de voltaje:
 - Comunicación básica: 2.0 – 3.6V
 - Otros comandos y acceso a memoria: 2.7 – 3.6V
- Rata de transferencia variable 0 – 25Mhz
- Tasa de transferencia hasta 12.5MB/sec (utilizando 4 líneas de datos)
- Máxima tasa de transferencia con hasta 10 tarjetas
- Corrección de errores de campo
- Mecanismo de protección de *Copyright*
- Protección con contraseña (algunas tarjetas)
- Mecanismos de protección contra escritura temporal y permanente

La especificación de durabilidad para cada sector de memoria es de 100.000 ciclos de escritura (la lectura de sector es ilimitada). Dicha capacidad excede el tiempo de vida útil de los dispositivos donde se utiliza la memoria (como celulares, multimedia players, cámaras, etc).

Como ilustración se puede mencionar que para malograr un sector de memoria se tendría que reescribir datos 3 veces por hora, durante 8 horas, los 365 días del año durante 10 años.

Las tarjetas SD soportan la “Inserción en Caliente” (*Hot Insertion*), es decir que a diferencia de las memorias flash USB, una expulsión por medio de software no es necesaria, ya que internamente la tarjeta posee circuitería de encendido y apagado de sistema automático; así como físicamente los electrodos de alimentación en el zócalo de la tarjeta son lo suficientemente extensos como para mantener la alimentación mientras los demás pines se conectan o desconectan adecuadamente.

Existe también la modalidad de bajo consumo de energía automático (*Sleep*) que se pone en funcionamiento cuando ningún comando ha sido

detectado desde el host luego de 5msec. El host no tiene que realizar ninguna acción para que esto ocurra, de la misma manera que para salir de esta modalidad. Estos procesos son autónomos en la memoria.

2.4.5 Registros principales

Existen varios registros asociados a la interfaz de la tarjeta de memoria, donde se guarda la información de configuración, protocolos de comunicación, y direccionamiento del sistema. A continuación se mencionan los principales:

Nombre	Tamaño en bits	Descripción
OCR	32	Registro de condiciones de operación
CID	128	Registro de Identificación
CSD	128	Registro de datos específicos
RCA	16	Registro de dirección relativa

Tabla 2.3 Registros principales de la tarjeta SD¹

Registro de condiciones de operación (OCR)

El registro OCR (*Operating Conditions Register*) es un registro de 32 bits donde se guarda la información del voltaje de trabajo de la tarjeta. Esta realiza un procedimiento de reconocimiento de voltaje de operación con el comando de inicialización CMD, detectando un rango de entre 2.0 y 3.6 voltios y almacenándolo en el registro.

Registro de identificación (CID)

El registro CID (*Card Identification*) es de longitud 16 bits y contiene un número único de identificación, programado durante la fabricación y no puede ser modificado.

(1) Nota: En el presente trabajo no se utilizarán ni modificarán estos registros.

Registro de datos específicos (CSD)

El registro CSD (*Card Specific Data*) contiene la información de configuración que se necesita para acceder a la tarjeta. El registro posee varios campos según sus características: estos pueden ser de solo lectura (R), de lectura y escritura (R/W), y de lectura / re-escritura (R/W / E). Algunos campos importantes son:

- Tiempo de acceso a lectura
- Rata de transferencia máxima
- Clases de comandos
- Longitud de bloque
- Parámetros para manejo de bloque múltiple
- Valores de corriente para lectura / escritura
- Tamaño de grupo de borrado
- Tamaño de sector de borrado
- Información de protección contra escritura, entre otros.

Registro de direcciones relativas (RCA)

El registro RCA (*Relative Card Address*) contiene la dirección con la que el host realiza el proceso de comunicación durante el procedimiento de inicialización. Este contenido es ignorado cuando la tarjeta se encuentra en modo SD.

2.4.6 Interfaz eléctrica

Configuración de pines

La configuración de los pines de la tarjeta y su función varían dependiendo de la modalidad de trabajo. A continuación se muestra la configuración en modalidad SD y SPI:

MODALIDAD SD

Pin #	Name	Type ¹	SD Description
1	CD/DAT3 ²	I/O ³	Card Detect/Data Line [Bit 3]
2	CMD	I/O	Command/Response
3	V _{SS1}	S	Supply voltage ground
4	V _{DD}	S	Supply voltage
5	CLK	I	Clock
6	V _{SS2}	S	Supply voltage ground
7	DAT0	I/O	Data Line [Bit 0]
8	DAT1	I/O	Data Line [Bit 1]
9	DAT2	I/O	Data Line [Bit 2]

Tabla 2.4 Configuración de terminales para el modo SD¹**MODALIDAD SPI**

Pin #	Name	Type ¹	SPI Description
1	CS	I	Chip Select (Active low)
2	DataIn	I	Host to Card Commands and Data
3	VSS1	S	Supply Voltage Ground
4	VDD	S	Supply Voltage
5	CLK	I	Clock
6	VSS2	S	Supply Voltage Ground
7	DataOut	O	Card to Host Data and Status
8	RSV(2)	I	Reserved
9	RSV(2)	I	Reserved

Tabla 2.5 Configuración de terminales para el modo SPI¹**2.4.7 Topología de bus**

La topología de bus nos da la idea de las modalidades en las que puede trabajar la tarjeta e interactuar con el host. A continuación se detallan:

Bus SD

En la modalidad SD el bus de comunicación posee seis líneas mientras que el bus de alimentación se compone de tres líneas.

(1) Fuente: <http://alumni.cs.ucr.edu/~amitra/sdcard/ProdManualSDCardv1.9.pdf>

- CMD – Línea de comando es una señal bidireccional.
- DAT [0:3] – Líneas de datos bidireccionales.
- CLK – Señal de reloj (*clock*) que proviene del host.
- VDD – Línea de alimentación positiva hacia todas las tarjetas.
- VSS [1:2] – Líneas de tierra o alimentación común.

En esta modalidad los comandos son enviados desde el host individualmente hacia cada tarjeta. La configuración para la utilización de múltiples líneas de datos se la realiza luego de la inicialización, al elegir la modalidad correcta.

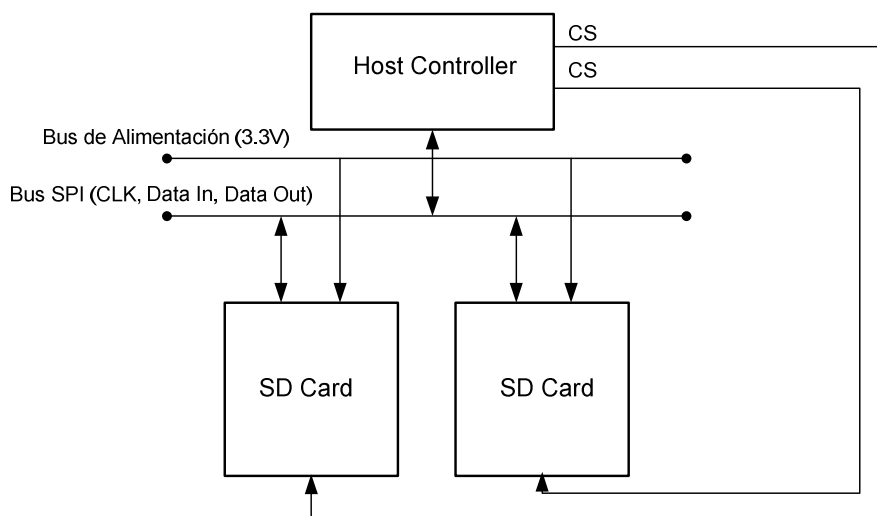
Bus SPI

En la modalidad SPI la tarjeta es compatible con las interfaces SPI disponibles en el mercado. Consiste en las siguientes cuatro líneas:

- CS – (*Chip Select*) señal de selección de tarjeta
- CLK (*Clock*) señal de reloj desde el host
- Data In – Señal de datos desde el host hacia la tarjeta
- Data Out – Señal de datos desde la tarjeta hacia el host

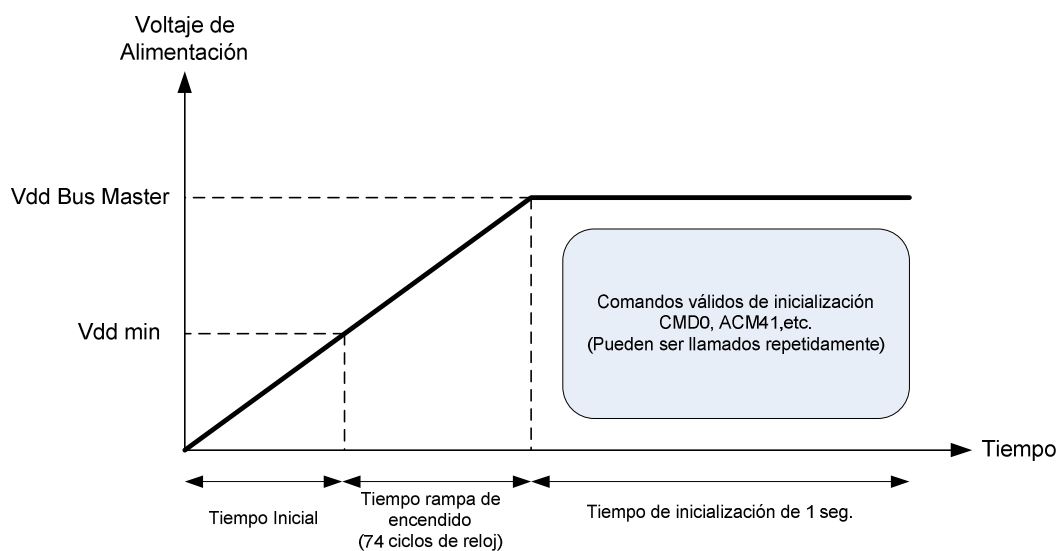
Los datos son transmitidos en unidades de 1 byte denominados testigos o tokens alineados a la señal CS. La implementación de identificación y direccionamiento en el modo SD es reemplazado por una señal de hardware CS para seleccionar la tarjeta en nivel bajo.

Las señales bidireccionales DAT son reemplazadas por líneas unidireccionales de datos. Esto elimina la posibilidad de recibir comandos mientras se realiza una operación de lectura / escritura.

Figura 2.14 Topología de bus SPI

2.4.8 Características eléctricas

Encendido

Figura 2.15 Diagrama de encendido (power up)

Después de conectar la alimentación de la tarjeta (incluyendo la inserción en caliente), ésta entra en un estado de espera (*IDLE*) y está preparada para realizar el proceso de inicialización. En el proceso de inicialización el host debe

iniciar el reloj durante algunos ciclos para lograr superar el tiempo de rampa de encendido (*Supply Ramp Up Time*) para lograr estabilizar el comportamiento de la tarjeta y proceder a la negociación del protocolo. El periodo máximo que dura el procedimiento de encendido es de 1 segundo.

Condiciones de operación de bus

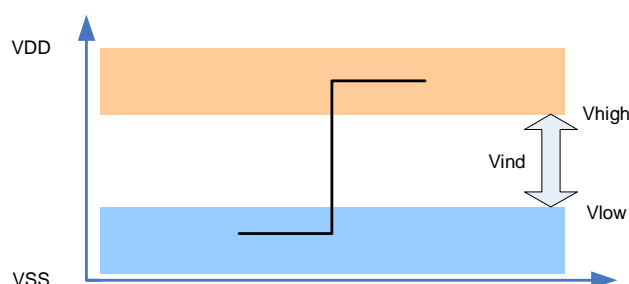
Las condiciones de operación eléctrica son idénticas en las dos modalidades. En la siguiente tabla se identifican las variables principales:

Variable	Min	Max	Unidad	Comentario
Voltaje pico en todas las líneas	-0,3	VDD+0,3	V	Para prevenir línea flotante $f \leq 20\text{Mhz}$ $f \leq 20\text{Mhz}$
Voltaje de alimentación	2,0	3,6	V	
Resistencia Pull-Up	10	100	K Ω	
Capacitancia máxima		10	pF	
Inductancia máxima		16	nH	

Tabla 2.6 Condiciones de operación de bus SD

Niveles de señal

Para cumplir con las especificaciones de voltaje de entrada y salida de la tarjeta SD, es necesario conocer el rango de trabajo de los voltajes de bus ubicados entre VDD y VSS dados por el siguiente gráfico.

Figura 2.16 Gráfico de niveles de voltaje en el bus de la tarjeta SD

VSS	Vlow	Vind	Vhigh	VDD
0V	0.825V	indefinido	2.475V	3.3V

Tabla 2.7 Niveles de voltaje en el bus SD

2.4.9 Protocolo de comunicación SPI

La comunicación entre la tarjeta de memoria y el host está basada en comandos y respuestas transmitidos mediante dos modalidades principales: Modo SD y modo SPI. En el presente caso se utiliza la modalidad SPI debido a que la interfaz se encuentra implementada en la mayoría de microcontroladores comerciales.

La comunicación SPI está basada en bloques de datos de 1 byte alineados a la señal CS (*chip select*). Los mensajes SPI están constituidos de comandos, respuestas y tokens de datos. Toda la transacción está comandada por el host (*master*) por medio de la señal CS en nivel bajo.

El comportamiento de la tarjeta en modo SPI difiere del modo SD en tres puntos:

- La tarjeta seleccionada siempre responde a un comando
- Una estructura de 8 ó 16 bits es utilizada
- Si la tarjeta encuentra un problema en los datos responderá con un error

Selección de modalidad

Cuando la tarjeta se enciende por defecto ésta se encuentra en modalidad SD. Entrará a modalidad SPI si solo si la señal de CS se activa durante la recepción del comando de *reset* (CMD0). Si la tarjeta reconoce la modalidad SPI responderá a través de una respuesta R1 (ver tokens de respuesta), caso contrario se mantendrá en modalidad SD y no responderá al comando.

El chequeo por medio de CRC (Comprobación de Redundancia Cíclica) en la modalidad SPI está deshabilitado por defecto (puede ser activado a voluntad por medio del comando específico), sin embargo en la selección de modalidad, la estructura del comando de inicialización debe contener los bytes de CRC respetando la estructura requerida por el protocolo.

Estructura de comandos SPI

Todos los comandos SPI contienen seis bytes de longitud transmitidos con MSB primero, donde se notan tres partes principales: los bits de inicio (siempre son '0' y '1'), el comando (1 byte), el argumento (4 bytes), el CRC (1 byte) y el bit de salida (siempre '1').

Figura 2.17 Estructura de los comandos SPI¹

Byte 1				Bytes 2—5				Byte 6	
7	6	5	0	31			0	7	0
0	1	Command		Command Argument				CRC	
									1

El código binario de un comando específico guarda relación con su mnemónico correspondiente. Por ejemplo el comando CMD0 en binario es '000000' y para el CMD39 el código es '100111'.

(1) Fuente: <http://alumni.cs.ucr.edu/~amitra/sdcard/ProdManualSDCardv1.9.pdf>

Comandos Principales

La tarjeta SD soporta dos tipos de comandos principales:

- Comandos regulares (Ejemplo: CMD0, CMD24, CMD17)
- Comandos de aplicación específica (Ejemplo: ACMD22, ACMD17)

Los comandos regulares están orientados a transacciones de información en sectores de datos de usuario y lectura de registros. Los comandos de aplicación específica se refieren al proceso de lectura y escritura de datos en sectores protegidos de la memoria así como modificación de registros especiales.

Cabe destacar que para invocar a un comando de aplicación específica, se requiere adicionar un comando previo (CMD55) en todas las llamadas para la identificación del comando específico. En el presente caso solo se utilizarán los comandos regulares.

CMD INDEX	Argumento	Resp.	Abreviación	Descripción de Comando
CMD0	Ninguno	R1	GO_IDLE_STATE	Reinicializa la tarjeta
CMD1	Ninguno	R1	SEND_OP_COND	Activa el proceso de inicialización
CMD9	Ninguno	R1	SEND_CSD	Informa a la tarjeta que debe enviar su información específica (CSD)
CMD10	Ninguno	R1	SEND_CID	Informa a la tarjeta que debe enviar su identificación (CID)
CMD13	Ninguno	R2	SEND_STATUS	Informa a la tarjeta que debe enviar su registro de estado
CMD16	[31:0] longitud de bloque	R1	en bytes	Establece la longitud de bloque (en bytes)
CMD17	[31:0] dirección de datos	R1	READ_SINGLE_BLOCK	Lee un bloque de datos de longitud establecida por SET_BLOCK_LEN
CMD24	[31:0] dirección de datos	R1	WRITE_BLOCK	Escribe un bloque de datos de longitud establecida por SET_BLOCK_LEN
CMD58	Ninguno	R3	READ_OCR	Lee el registro OCR
CMD59	[31:1] No importa [0:0] Opción CRC	R1	CRC_ON_OFF	Controla el encendido o apagado de la opción CRC (1: ON y 0: OFF)

Tabla 2.8 Resumen de comandos principales de tarjeta SD

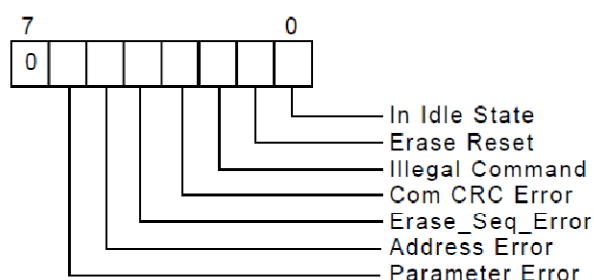
TOKENS DE RESPUESTA Y TIPOS DE FORMATO

Existen distintos tipos de tokens de respuesta. Estos bytes son transmitidos con el bit MSB primero. El tipo de respuesta depende de la operación realizada en la tarjeta y se detalla a continuación.

Formato R1

Este byte de respuesta es enviado por la tarjeta SD después de cada comando con excepción del comando *SEND_STATUS*. Tiene una longitud de 1 byte, el bit MSB es siempre cero y los demás bits indican el tipo de error cuando su estado es uno lógico.

Figura 2.18 Estructura de respuesta R1¹



- *In Idle State* – La tarjeta se encuentra en modo *IDLE* durante el proceso de inicialización (ver capítulo de diseño de software de control).
- *Erase Reset* - Indica que una secuencia de borrado fue cancelada antes de ser ejecutada debido a que se ha recibido un comando de salida del proceso.
- *Illegal Command* – Indica que un código erróneo de comando ha sido detectado.
- *Communication CRC Error* – El chequeo de CRC del último comando ha fallado.
- *Erase Sequence Error* – Ha ocurrido un error en la secuencia de comandos de borrado.

(1) Fuente: <http://alumni.cs.ucr.edu/~amitra/sdcard/ProdManualSDCardv1.9.pdf>

- *Address Error* – Una dirección fuera del rango, que no coincide dentro de la longitud de bloque establecida ha sido detectada.
- *Parameter Error* – El argumento del comando se encuentra fuera de rango de la tarjeta.

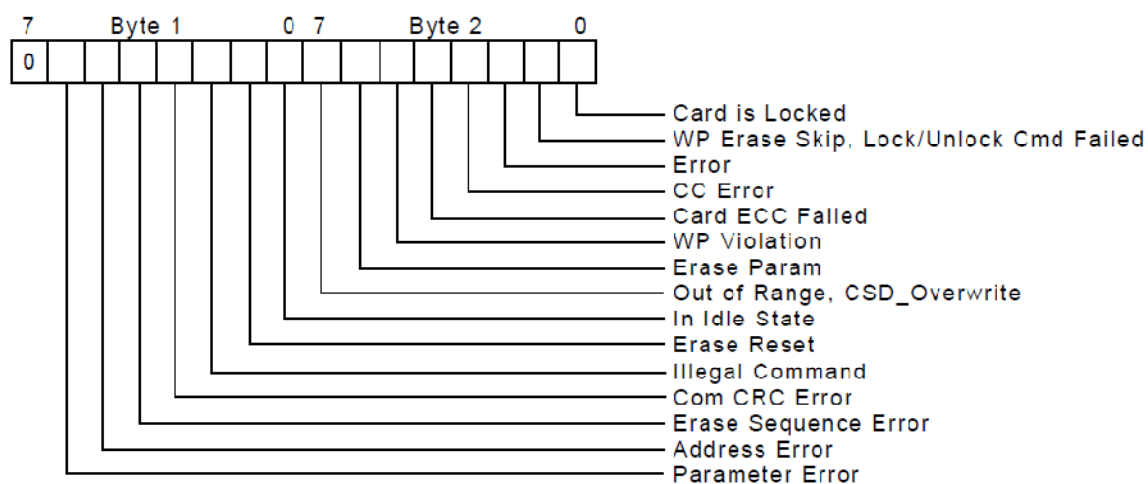
Formato R1b

Este byte de respuesta es idéntico al formato R1 excepto que la señal de ocupado (*busy signal token*) está adicionada el byte R1 de una longitud indefinida con valor de cero. Cuando un valor diferente a cero es recibido por el host entonces la tarjeta está lista para el siguiente comando.

Formato R2

Esta respuesta es de 2 bytes de longitud y es enviada por la tarjeta al responder al comando *SEND_STATUS*. El formato es el siguiente:

Figura 2.19 Estructura de respuesta R2¹



El primer byte es idéntico a R1. El contenido del segundo byte es el siguiente:

- *Erase Param* – Selección de sectores de borrado inválida.

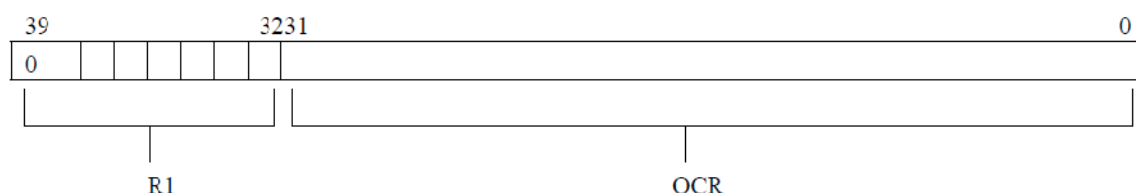
(1) Fuente: <http://alumni.cs.ucr.edu/~amitra/sdcard/ProdManualSDCardv1.9.pdf>

- *Write Protect Violation* – El comando intentó escribir un bloque protegido.
- *Card ECC Failed* – ECC interno fué aplicado pero falló en corregir datos.
- *CC error* – Error de controlador interno.
- *Error* – Ha ocurrido un error general de característica desconocida.
- *Write Pprotect Erase Skip* - Se ha borrado parcialmente un espacio seccionado.
- *Card Is Locked* - La tarjeta está en modo bloqueado (solo tarjetas Sandisk).

Formato R3

Esta respuesta es enviada por la tarjeta SD solo si el comando *READ_OCR* ha sido recibido. La respuesta es de longitud 5 bytes. La estructura del primer byte (MSB) es idéntica a la respuesta R1. Los otros 4 bytes contienen el registro OCR.

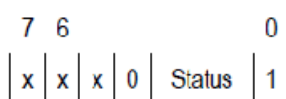
Figura 2.20 Estructura de respuesta R3¹



Token de respuesta de datos

Cada bloque de datos escritos en la tarjeta están confirmados por un token de respuesta de datos. El token tiene un byte de longitud con el siguiente formato:

Figura 2.21 Estructura de byte de respuesta de datos¹



El significado de los bits de estado (*status bits*) se define como:

(1) Fuente: <http://alumni.cs.ucr.edu/~amitra/sdcard/ProdManualSDCardv1.9.pdf>

- '010'—Los datos fueron aceptados.
- '101'—Los datos fueron rechazados debido a un error de CRC.
- '110'—Los datos fueron rechazados debido a un error de escritura.

En el caso de cualquier error de datos causados por CRC o escritura durante una operación de escritura de bloque simple o múltiple, el host debe detener el proceso de transmisión usando CMD12. En el caso de error de escritura (respuesta '110') el host debe enviar CMD13 (*SEND_STATUS*).

Token de datos

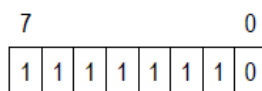
Los comandos de lectura y escritura tienen transferencia de datos asociados. Los datos son recibidos y transmitidos mediante token de datos. Todos los bytes son transmitidos mediante MSB primero.

Los tokens de datos son de 4 hasta 515 bytes de longitud y tienen el siguiente formato:

Para una operación de lectura de bloque simple, escritura de bloque simple y lectura de bloque múltiple:

- Primer Byte: Bloque de Inicio.

Figura 2.22 Estructura de token de datos



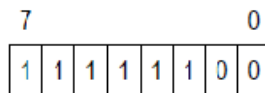
- Bytes 2 al 513 (dependiendo de la longitud del bloque): datos de usuario.
- Últimos dos bytes: CRC de 16-bits.

Para una operación de escritura de bloque múltiple:

- Primer byte de cada bloque.

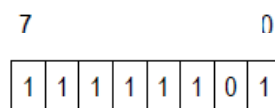
Si los datos van a ser transmitidos entonces el primer bloque es:

Figura 2.23 Estructura de token de datos 2



Si una cancelación de transmisión es requerida:

Figura 2.24 Estructura de token de datos 3

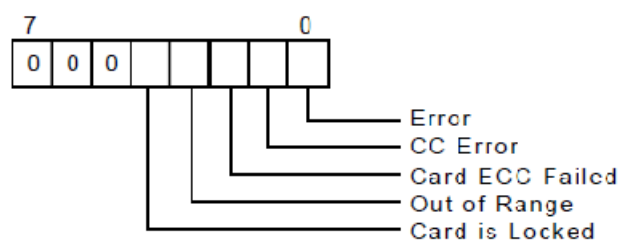


Se debe notar que esta estructura es solo para una operación de escritura de bloque múltiple. En el caso de solicitar una parada durante una operación de lectura de bloque múltiple, se debe usar el comando *STOP_TRAN* (CMD12).

Token de error de datos

Si una operación de lectura de datos falla y la tarjeta es incapaz de proveer la información solicitada, un token de error es entonces enviado en su lugar. Este token es de longitud 1 byte y tiene el siguiente formato:

Figura 2.25 Estructura de token de error de datos¹



(2) Fuente: <http://alumni.cs.ucr.edu/~amitra/sdcard/ProdManualSDCardv1.9.pdf>

Los últimos bits menos significativos (LSB) son los mismos bits de error de la respuesta de tipo R2.

2.4.10 Características de transferencia de datos

En la siguiente tabla se muestra el detalle de las tasas de transferencia máximas y velocidades de reloj entre las tarjetas Multimedia y Secure Digital, en sus diferentes modalidades de trabajo.

Product		Maximum Clock Speed and Burst Rate	
MultiMediaCard		Clock Speed	Burst Rate
SPI Bus mode	20 MHz		2.5 MB/s
MMC 1-bit mode	20 MHz		2.5 MB/s
SD Card			
SPI Bus mode	25 MHz		3.125 MB/s
SD 1-bit mode	25 MHz		3.125 MB/s
SD 4-bit mode	25 MHz		12.5 MB/s

Tabla 2.9 Comparación de tasas de datos de tarjetas MMC y SD¹

2.4.11 Nuevas tecnologías en tarjetas SD

La tecnología de miniaturización ha permitido que las memorias SD incorporen circuitería inalámbrica en su estructura interna, logrando establecer un enlace con dispositivos Wi-Fi en una red inalámbrica casera. Esta tecnología de Wi-Fi sobre SD card se denomina Eye-Fi.

Figura 2.26 Tarjeta de memoria SD con tecnología inalámbrica Eye-Fi²



(1) Fuente: sandisk product manual

(2) Fuente: www.amazon.com/Eye-Fi-Card-Wireless-GB-Memory/dp/B000X27XDC

2.5 Pantalla LCD de gráficos (GLCD)

Las pantallas LCD de gráficos ó GLCD son dispositivos visuales capaces de mostrar gráficos basados en una matriz de puntos donde cada punto se relaciona con un mapa de bits en una memoria RAM de gráficos interna (DDRAM). La resolución de la pantalla está en relación directa con el número de puntos por unidad de longitud que es capaz de mostrar, así como también de la resolución depende el volumen de datos que maneja la pantalla y su controlador.

Figura 2.27 Vista frontal del GLCD TS12864B



La pantalla utilizada es de tipo TS12864B de 128 x 64 puntos, y está dividida en dos partes, es decir en dos pseudo-pantallas de 64 x 64 que son manejadas por dos controladores idénticos tipo KS0108 de Samsung. Cada controlador se direcciona independientemente y es capaz de manejar ocho páginas horizontales de 8 bits (1 byte), es decir el controlador contiene la memoria RAM dedicada a gráficos de 64 x 64 bits, lógica de decodificación y latches de 64 bits. En la figura 2.28 se observa la distribución de las páginas de memoria de gráficos en el GLCD.

La circuitería de la pantalla trabaja con el controlador en común tipo KS0107 que provee de señales de temporización para los controladores KS0108, registros de desplazamiento y drivers de 64 bits.

Figura 2.28 Distribución de memoria en la pantalla del GLCD

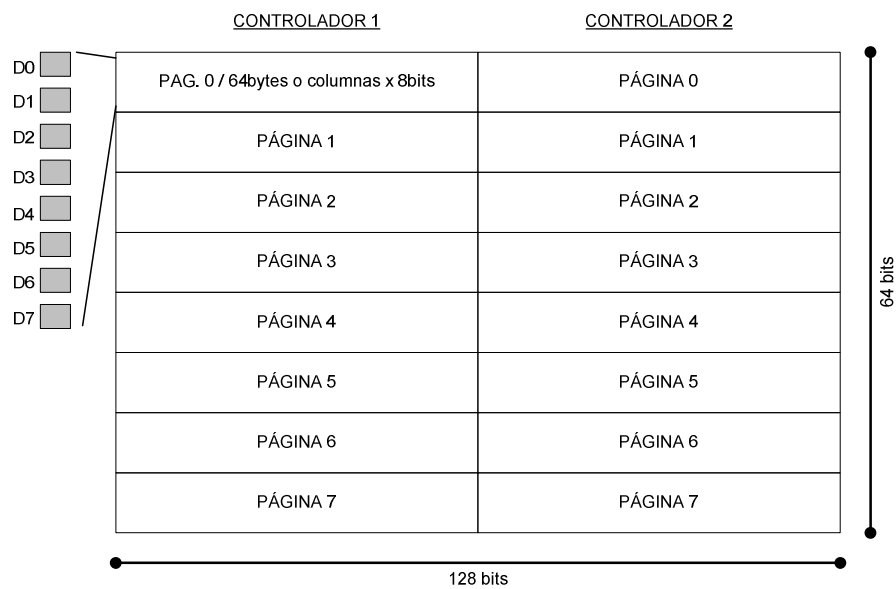
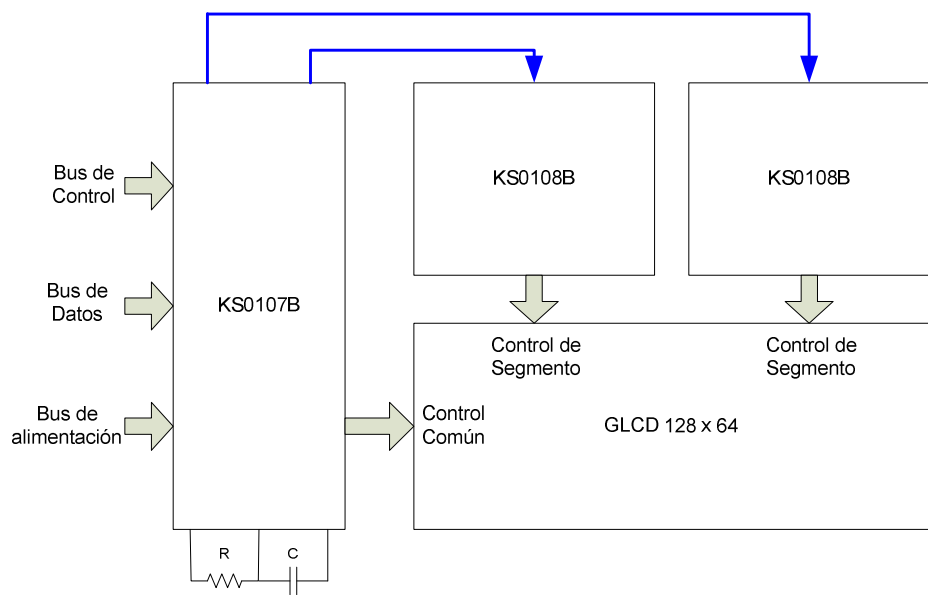


Figura 2.29 Diagrama de bloques de la estructura interna del GLCD



La comunicación del controlador con el dispositivo gráfico es paralela y consta de tres buses: bus de datos, bus de control, bus de alimentación.

El bus de alimentación consta de la línea principal de 5V de donde se polariza toda la circuitería interna. Se dispone también de una línea de

salida de voltaje negativo (VEE), de alrededor de - 20V que genera el mismo LCD a través de un conversor DC/DC para controlar el contraste de la pantalla. Existen también las líneas de alimentación del arreglo de leds de *backlight* o luz de fondo (Led A y Led K) de 5V.

El bus de control permite la identificación de tipo de dato, tipo de operación, control de validación, y selección de controlador interno; como son: la línea D/I (*data / instruction*), la línea R/W (*read / write*), la línea E (*enable*) y las líneas CS1 y CS2 (*chip select*) respectivamente.

El bus de datos consta de 8 líneas paralelas, donde viajarán los datos y las instrucciones como palabras de 1 byte. A continuación se muestra el detalle de configuración de pines del GLCD:

Pin NO.	Symbol	Function	Pin NO.	Symbol	Function
1	Vss	GND	11	DB4	Data bus line
2	Vdd	Power supply (+5V)	12	DB5	Data bus line
3	Vo	Contrast Adjustment	13	DB6	Data bus line
4	D/I	Data/instruction	14	DB7	Data bus line
5	R/W	Data read/write	15	CS1	Chip select for IC1
6	E	H→L Enable signal	16	CS2	Chip select for IC2
7	DB0	Data bus line	17	RST	Reset
8	DB1	Data bus line	18	Vee	Negative voltage output
9	DB2	Data bus line	19	A	Power supply for LED+(4.2V) RA=0Ω
10	DB3	Data bus line	20	K	Power supply for LED (0V)

Tabla 2.10 Distribución de pines del GLCD TS12864B¹

2.6 MICROCONTROLADORES: RECURSOS DE COMUNICACIÓN Y HERRAMIENTAS DE DESARROLLO.

La principal ventaja del uso de microcontroladores es su relación costo-efectividad; pues por un precio asequible, se obtiene una herramienta muy completa para el diseño digital. Existen un sinnúmero de aplicaciones que van en relación con las características internas del microcontrolador, como longitud de palabra (4, 8, 16, 32 y 64 bits), cantidad de operaciones por segundo

(1) Fuente: <http://www.lcdproduct.com/a/ensystem/pdf/URL/PGM128128A.pdf>

(MIPS), periféricos de Entrada/Salida, prestaciones especiales (puertos SPI, I2C, Ethernet, USB, Blue Tooth), etc.

Figura 2.30 Ejemplo de microcontrolador: MC68HC11 de Motorola¹



2.6.1 Microcontrolador PIC18F4550

Al momento de elegir el microcontrolador para la implementación del proyecto, se debieron tomar en cuenta varios factores que incidieron en la toma de decisión. En orden de importancia estos son:

- Disponibilidad en el mercado
- Relación costo-prestaciones
- Tipo de encapsulado dependiendo de la tecnología disponible
- Velocidad de procesamiento y frecuencia de reloj
- Cantidad de memoria de programa (FLASH)
- Cantidad de memoria de procesamiento (RAM)
- Cantidad de líneas de E/S necesarias para la aplicación
- Prestaciones especiales
- Disponibilidad de software de desarrollo
- Disponibilidad de programador de microcontrolador (grabador)

En el presente caso se analizaron los puntos anteriores y se llegó a la conclusión de utilizar el microcontrolador PIC18F4550 de la casa Microchip[®], debido a que posee las prestaciones necesarias, además existe en el mercado local y es el más rápido disponible con empaque DIP. Posee también mayor

(1) Fuente: http://es.wikipedia.org/wiki/Archivo:MC68HC11_microcontroller.jpg

cantidad de memoria de programa (en relación con los más comunes), tiene un costo relativamente bajo, y se dispone de software/hardware para desarrollo.

En los capítulos 5 y 6, se especifican los parámetros internos y la configuración del microcontrolador; así como también se detalla la asignación de las líneas de Entrada/Salida.

En la siguiente tabla se mencionan algunas de las características sobresalientes del microcontrolador:

Microchip® PIC18F4550	
Memoria de Programa (flash)	32 Kbytes
Frecuencia de Reloj Máxima	48Mhz
Memoria RAM interna	2048 bytes
Memoria Eprom interna	256 bytes
Número de canales ADC	13 (10-bit de resolución)
Número de timers de 8 bits	1
Número de timers de 16 bits	3
Número de entradas/salidas	35
Varios: Comunicación serial SPI, I ² C, entre otros.	

TABLA 2.11 Principales características del microcontrolador PIC18F4550

2.6.2 Comunicación serial RS-232

RS-232 (también conocido como *Electronic Industries Alliance RS-232C*) es una interfaz que designa una norma para el intercambio serie de datos binarios entre un DTE (Equipo Terminal de Datos) y un DCE (Equipo de Comunicación de Datos).

Características básicas

1. Características eléctricas de la señal: Se establece que la longitud máxima entre el DTE y el DCE no debe ser superior a los 15 metros. Los niveles lógicos son:
 - a. 1 lógico entre -3V y -15V
 - b. 0 lógico entre +3V y +15V

2. Características mecánicas de los conectores: Se utiliza un conector DB25, o DB9, donde el conector macho identifica al DTE y el conector hembra al DCE.

3. Descripción funcional de las señales usadas: Las señales están básicamente divididas en dos grupos:
 - a. Señales primarias, que son normalmente utilizadas para la transferencias de datos
 - b. Señales secundarias, utilizadas para el control de la información que será transferida.

La norma RS232 está definida tanto para la transmisión síncrona como para la asíncrona, pero cuando se utiliza esta última, sólo el conjunto de terminales DB25, es utilizado.

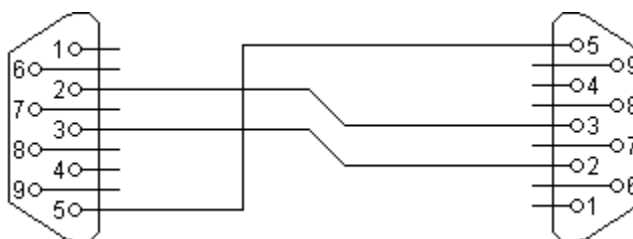
A continuación se detallan la asignación de pines para los conectores en RS232. Tanto DTE como DCE son los mismos, sin embargo en la tabla siguiente se observa desde DTE (computador).

Numero En DB-25	de Pin En DB-9	Señal	Descripción	E/S
1	1	-	Masa chasis	-
2	3	TxD	Transmit Data	S
3	2	RxD	Receive Data	E
4	7	RTS	Request To Send	S
5	8	CTS	Clear To Send	E
6	6	DSR	Data Set Ready	E
7	5	SG	Signal Ground	-
8	1	CD/DCD	(Data) Carrier Detect	E
15	-	TxC(*)	Transmit Clock	S
17	-	RxC(*)	Receive Clock	E
20	4	DTR	Data Terminal Ready	S
22	9	RI	Ring Indicator	E
24	-	RTxC(*)	Transmit/Receive Clock	S

TABLA 2.12 Descripción de pines en los conectores usados en RS232

En la comunicación asíncrona básica sin control de flujo por hardware, la conexión RS232 se establece solo con tres líneas: RXD, TXD y GND, como se indica en el siguiente gráfico:

Figura 2.31 Conexión RS232 asíncrona

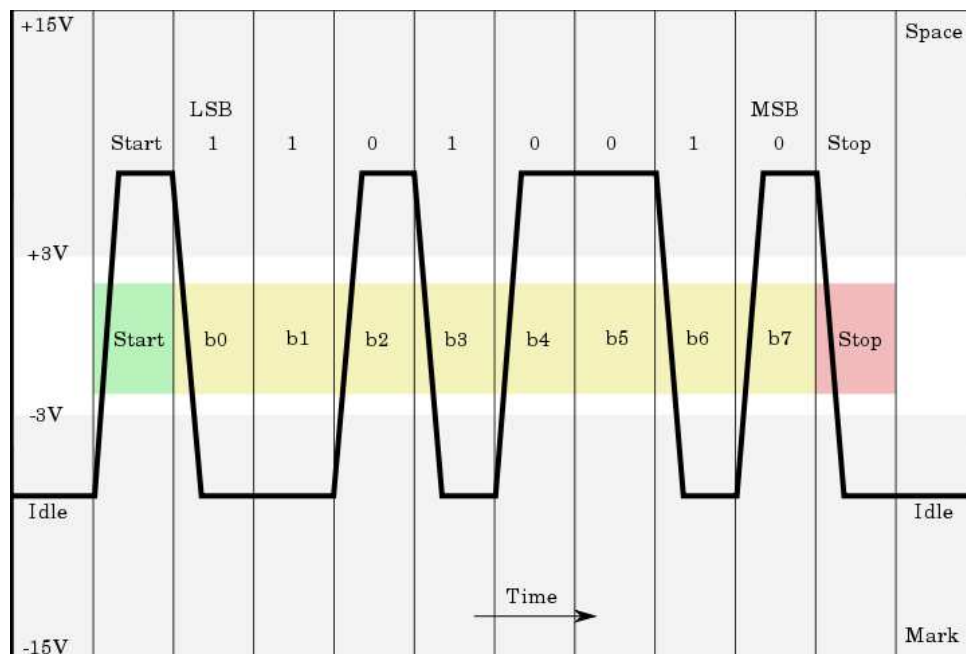


Estructura de señal

La señal RS232 está formada por un bit de partida (*Start Bit*), los siguientes ocho bits son la palabra ó dato con LSB primero, luego el bit de paridad (opcional), y luego los bits de parada (*Stop Bit*) que dependiendo de la

configuración pueden ser de 1 a 2. En el siguiente gráfico se muestra el dato 11010010 siendo transmitido.

Figura 2.32 Estructura de la señal RS232¹



Adaptación de niveles

Debido a que el microcontrolador maneja niveles TTL y lógica inversa en la comunicación serial, es necesario incluir un circuito conversor de niveles de voltaje para manejar un puerto serial RS232 en el prototipo y permitir la intercomunicación con el receptor GPS y la PC.

Se utiliza entonces un par de transceptores del conocido integrado MAX232 cuya configuración se detalla en el capítulo 5.

2.6.3 Comunicación serial SPI

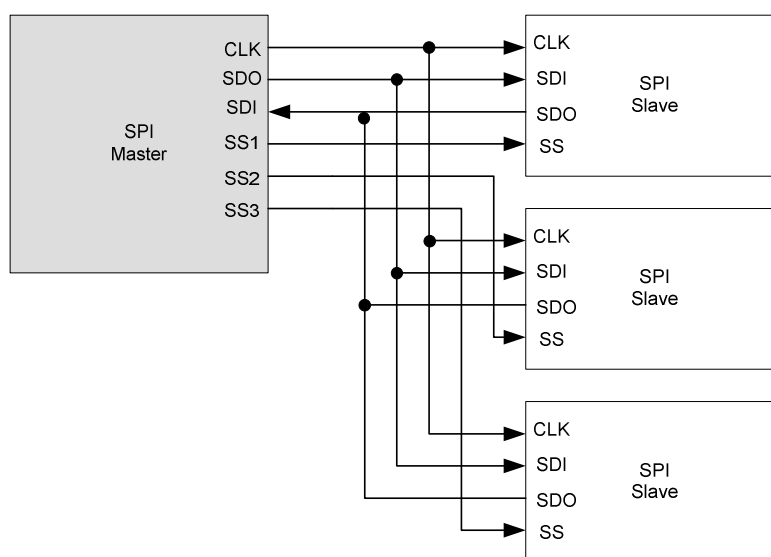
El Bus SPI (*Serial Peripheral Interface*) es un estándar de comunicaciones, usado principalmente para la transferencia de información

(1) Fuente: http://en.wikipedia.org/wiki/File:Rs232_oscilloscope_trace.svg

entre circuitos integrados en equipos electrónicos. El bus de interface de periféricos serie o bus SPI permite la comunicación digital por medio de un bus de datos y un reloj compartido necesitando una línea adicional para direccionar un dispositivo.

Existe siempre un dispositivo maestro (*master*), y el comanda a los demás dispositivos esclavos (*slaves*). El dispositivo *master* controla la generación del reloj y direcciona los dispositivos en el bus de comunicación.

Figura 2.33 Ejemplo de configuración de bus SPI



La nomenclatura utilizada corresponde a:

- CLK — (Clock): Reloj de sincronización generado por el *Master*.
- SDO — (Serial Data Out): Salida Serial de Datos desde el *Master* ó desde los *Slaves* hacia el bus SPI.
- SDI — (Serial Data Input): Entrada Serial de Datos hacia el *Master* ó hacia los *Slaves* desde el bus SPI.
- SS — (Slave Select): Selección de Dispositivo Esclavo de activación negativa (bajo nivel) que permite direccionar los *Slaves*.

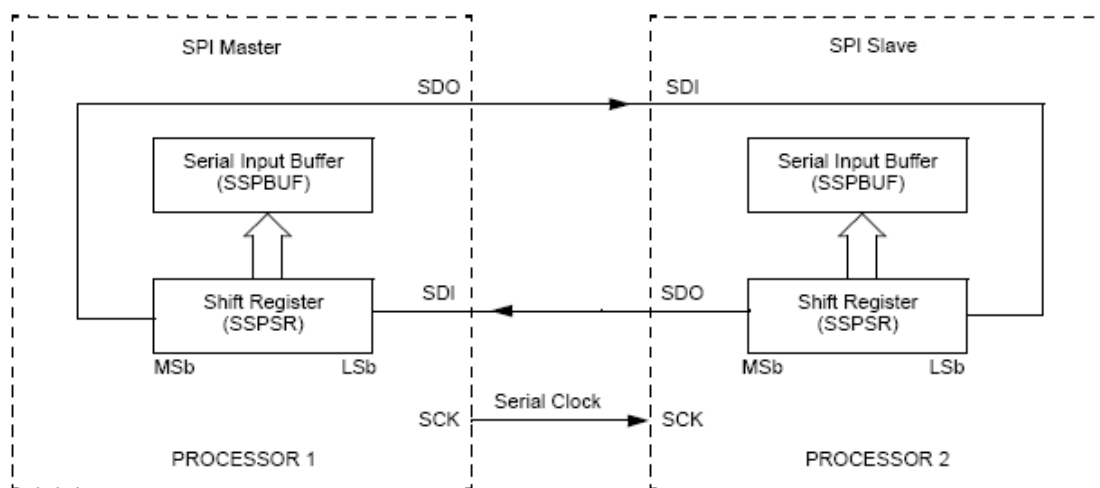
Bus SPI en el microcontrolador PIC18F4550

El bus SPI en el microcontrolador es comandado por el módulo MSSP (Master Synchronous Serial Port) ó Puerto Serial Sincrónico Maestro; que puede configurarse de dos formas: I2C y SPI.

En la modalidad SPI, intervienen tres registros en el control y estado del bus. Estos registros son: SSPBUF, SSPSR, SSPCON1 y SSPSTAT.

1. Registro SSPBUF .- Es un registro de lectura / escritura y constituye el registro *buffer* de transmisión y recepción de datos SPI.
2. Registro SSPSR .- Es un registro de desplazamiento que forma parte de la cadea de recepción/transmisión del bus SPI y no es accesible directamente.
3. Registro SSPCON1 .- Es un registro de control de bus SPI.
4. Registro SSPSTAT .- Es un registro de estado y control de bus SPI.

Figura 2.34 Diagrama de conexión Master/Slave del bus SPI¹



Se debe recalcar que el microcontrolador actuará como Master para gestionar la comunicación del bus SPI.

(1) Fuente: <http://ww1.microchip.com/downloads/en/DeviceDoc/39632b.pdf>

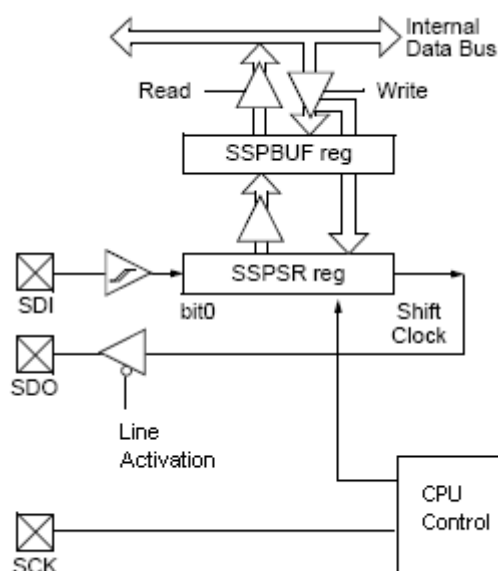
Transmisión de datos

Para la transmisión de una palabra ó byte por el bus SPI, el controlador coloca el dato que se encuentra en el bus principal del microcontrolador, hacia el registro búfer SSPBUF, y a su vez éste byte es copiado al registro de desplazamiento SSPSR para su transmisión serial. Bit a bit, con cada ciclo de reloj SCK generado en el microcontrolador, el byte es transmitido.

Recepción de datos

En la recepción los datos seriales vienen del exterior alojándose en el registro de desplazamiento SSPSR. Cuando el registro SSPSR está lleno, el bit 0 del registro de control SSPSTAT se activa indicando que la recepción está completa. En el momento que el búfer SSPSR se llena el byte es copiado hacia SSPBUF para poder ser leído en el bus de datos del microcontrolador.

Figura 2.35 Diagrama de bloques interno del controlador SPI¹



En la figura 2.35 se observa la conexión interna entre los componentes principales del controlador SPI en el microcontrolador. La entrada de datos SDI

(1) Fuente: <http://ww1.microchip.com/downloads/en/DeviceDoc/39632b.pdf>

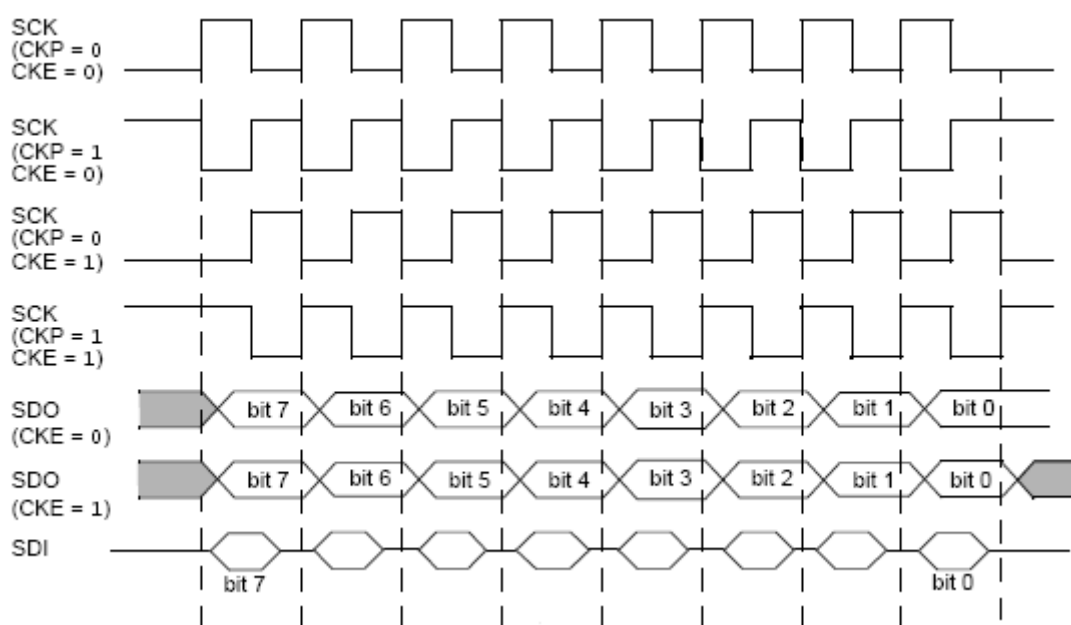
tiene un circuito trigger o disparador de histéresis, que actúa como amplificador de señal digital. En la salida de datos SDO, existe el habilitador de puerto que sirve para controlar la salida de datos ó no hacia otro dispositivo SPI. Deshabilitando esta línea permite enviar ciclos de reloj sin datos para sincronizar ó inicializar dispositivos externos como la memoria SD en este caso.

Temporización

En los registros de SSPCON1 Y SSPSTAT se configuran los dos parámetros principales de funcionamiento del bus SPI. El parámetro CKE indica que la transmisión se realiza en la transición de ciclo de reloj positiva o negativa, y el parámetro CKP indica el nivel de descanso del reloj (*Idle State*).

- CKE: Bit de Configuración de Reloj
1 = La transmisión ocurre en la transición de activo a pasivo (*Idle*)
0 = La transmisión ocurre en la transición de pasivo (*Idle*) a activo
- CKP: Bit de Selección de Polaridad de Reloj
1 = El estado pasivo (*Idle*) es un nivel alto
0 = El estado pasivo (*Idle*) es un nivel bajo

Figura 2.36 Diagrama de temporización del bus SPI¹



(1) Fuente: <http://ww1.microchip.com/downloads/en/DeviceDoc/39632b.pdf>

2.6.4 Protocolo de transmisión de datos Xmodem

Desde el punto de vista del hardware, Xmodem es implementado sobre líneas seriales asincrónicas, con configuración de 8 bits de datos, sin bit de paridad y un bit de parada. Xmodem utiliza los siguientes caracteres ASCII especiales:

Nombre	Decimal	Hexadecimal	Descripción
SOH	01	H01	Comienzo Encabezado
EOT	04	H04	Fin de la transmisión
ACK	06	H06	Confirmación (positiva)
DEL	16	H10	Escape
X-On (DC1)	17	H11	Transmisión On
X-Off (DC3)	19	H13	Transmisión Off
NAK	21	H15	Confirmación(negativa)
SYN	22	H16	Synchronous idle
CAN	24	H18	Cancelar

Tabla 2.13 Caracteres especiales en el protocolo Xmodem

Se considera como transmisor la computadora/software que envía paquetes y recibe confirmaciones y como receptora la computadora/software que recibe paquetes y envía confirmaciones. Xmodem es un protocolo que no incluye nombre de archivo en sus paquetes por lo que previamente, ambos dispositivos, transmisor y receptor, deben conocer donde encontrar los datos (que archivo será transmitido) y donde poner los datos (archivo que almacenara los datos o buffer).

Una vez entrado en el protocolo, el computador transmisor, espera entre 10 segundos y un minuto, recibir un carácter NAK de la computadora receptora. Se dice entonces, que la computadora receptora controla el manejo del protocolo. La computadora transmisora puede repetir esto cualquier número de veces. Si un carácter diferente a NAK o CAN es recibido por el transmisor, este lo ignora. El carácter CAN implica la cancelación de la transferencia de archivo

y el transmisor debe salir del protocolo Xmodem. Una vez que el receptor ha enviado un carácter NAK, espera por 10 segundos la llegada del primer paquete Xmodem. Si no recibe nada en estos 10 segundos, envía otro NAK, repitiendo esto por 10 veces, luego de lo cual de no haber recibido respuestas, sale del protocolo Xmodem, indicando un error severo.

El transmisor toma los datos y los divide en piezas de 128 bytes, y los coloca en un paquete Xmodem. Un paquete Xmodem consta de:

[SOH] [seq] [cmpl seq] [128 bytes de datos] [csum]

donde:

- **SOH:** Es un carácter de comienzo de cabecera
- **seq:** Un byte de número de secuencia que comienza en 1, incrementándose en uno hasta alcanzar el valor de 255
- **cmpl seq:** Byte de complemento a uno de seq
- **datos:** 128 bytes de datos
- **csum:** Byte de Checksum que representa la suma exclusiva lógica de todos los datos excepto los tres primeros del encabezado

Una vez que la inicialización del protocolo ha sido completada, el transmisor envía el primer paquete Xmodem y entonces espera. Luego que el receptor tiene el paquete completo, este compara su propio checksum calculado con el checksum que fue enviado en el paquete. Si los checksum coinciden, el receptor envía una confirmación positiva (ACK). Si los checksum son diferentes, el receptor envía una confirmación negativa (NACK). Después que el receptor envía un ACK, el transmisor envía el siguiente paquete Xmodem.

Si el transmisor recibe un NAK, este reenvía nuevamente el mismo paquete Xmodem. Una vez que el transmisor ha enviado el último paquete y ha recibido la confirmación positiva (ACK), respectivamente, este envía un EOT al receptor y entonces espera por un ACK final antes de salir de Xmodem.

Cuando el receptor ve un carácter EOT en lugar de un SOH, el receptor transmite un ACK y cierra sus archivos saliendo del protocolo Xmodem. Todo lo anterior esta representado en la siguiente gráfica:

Figura 2.37 Secuencia de transmisión de datos en Xmodem¹

Transmisor		Receptor
	<<<<<<<<	[NAK]
[SOH] [001] [254] [.....] [csum]	>>>>>>>>	
	<<<<<<<<	[ACK]
[SOH] [002] [253] [.....] [csum]	>>>>>>>>	
	<<<<<<<<	[ACK]
[SOH] [003] [252] [.....] [csum]	>>>>>>>>	
	<<<<<<<<	[ACK]
[SOH] [004] [251] [.....] [csum]	>>>>>>>>	
	<<<<<<<<	[ACK]
[EOT]	>>>>>>>>	
	<<<<<<<<	[ACK]

El receptor puede cancelar la transferencia enviando un carácter CAN y saliendo del protocolo. Si el transmisor recibe un carácter CAN cuando espera un carácter ACK o NAK, o cuando espera un carácter SOH (comienzo de paquete), este termina la transferencia y sale del protocolo. Muchas implementaciones del protocolo Xmodem, requieren de dos caracteres CAN ante de reconocer una condición de cancelación.

2.6.5 Herramientas de desarrollo

La disponibilidad de software de herramientas de desarrollo para microcontroladores es muy amplia. Existe una gama extensa de compiladores en distintos lenguajes como Ensamblador, C, Basic, Visuales, entre otros. La gran demanda existente ha permitido incluso la aparición en el mercado de tarjetas de desarrollo para microcontroladores de diversas marcas y de aplicaciones múltiples.

(1) Fuente: <http://www.bolivar.udo.edu.ve/microinternet/articulos/Xmodem.pdf>

Debido al tipo de aplicaciones cada vez más complejas, los lenguajes de alto nivel son utilizados con más frecuencia por incorporar herramientas de control y manejo de periféricos más completas y sencillas de usar. Estas aplicaciones serían mucho más extensas de desarrollar con lenguajes de bajo nivel como ensamblador.

En el presente caso se utiliza el compilador CCS que utiliza el lenguaje C / C++ de programación para desarrollar el prototipo de navegación GNOM-4550.

CAPÍTULO 3

DISEÑO DE FORMATO DE ARCHIVO Y SISTEMA DE ARCHIVOS

C O N T E N I D O

-
- 3.1 Diseño del nuevo formato de archivo de mapa vectorial**
 - 3.2 Diseño del nuevo sistema de archivos para memoria SD**
 - 3.3 Gestión del sistema**
 - 3.4 Resumen de características**
-

3.1 DISEÑO DEL NUEVO FORMATO DE ARCHIVO DE MAPA VECTORIAL

3.1.1 Introducción

El nuevo formato de archivo que se plantea, forma parte de el tipo de archivo mencionado en el marco teórico denominado como “mapa vectorial”, que es un tipo de mapa digital que contiene información acerca de formas geométricas básicas que en términos de coordenadas, forman un mapa en su conjunto, con información adicional de texto y georeferencia.

El formato nuevo formato de archivo pretende encapsular en su interior información de navegación que la unidad prototipo GNOM-4550 será capaz de leer e interpretar, presentando dicha información en pantalla y relacionándola con la posición actual del usuario mediante los datos que vienen del receptor GPS conectado al prototipo GNOM-4550.

Se pretende utilizar el concepto de capas o *layers* que permite la discriminación ó no de información por grupos para ser presentados en la pantalla. Por ejemplo en una primera capa pueden encontrarse datos de calles principales de una ciudad, en una segunda capa; datos de calles secundarias, en una tercera; líneas de tren, en una cuarta; ríos, etc.

El formato de archivo planteado se lo ha denominado GMV (Geo-Referenced Microcontroller Vector Map) ó mapa vectorial geo-referenciado para microcontrolador. La extensión de archivo es “gmv”.

3.1.2 Características generales

La complejidad del en la estructura del archivo debe ir de la mano con la capacidad de manejo de datos del microcontrolador. Se busca entonces que el sistema de coordenadas utilizado en un archivo sea sencillo; esto significa

utilizar un mínimo de 1 byte como unidad de información básica, salvo en casos especiales.

En términos de simplicidad para el software gráfico, el desplazamiento de la pantalla a través del mapa se lo realiza en dos ejes de coordenadas. Los ejes de coordenadas son el longitudinal y el latitudinal con su centro en la intersección del meridiano cero y el ecuador; esto es en el mar atlántico cerca a África central.

Para facilitar el trabajo del controlador del sistema, los archivos se los considera de cobertura geográfica constante; es decir cada archivo puede tener tanta información como sea necesario, pero siempre dentro de un área geográfica definida y constante. De esta manera el controlador armará el conjunto de archivos necesarios para la navegación según el área solicitada.

La cantidad de información contenida en un archivo de navegación es indefinida; la limitante en teoría es el área geográfica que define el archivo. La limitante del contenido de navegación en la práctica es la respuesta y la velocidad del sistema donde se interpreta el archivo; en este caso el microcontrolador del prototipo.

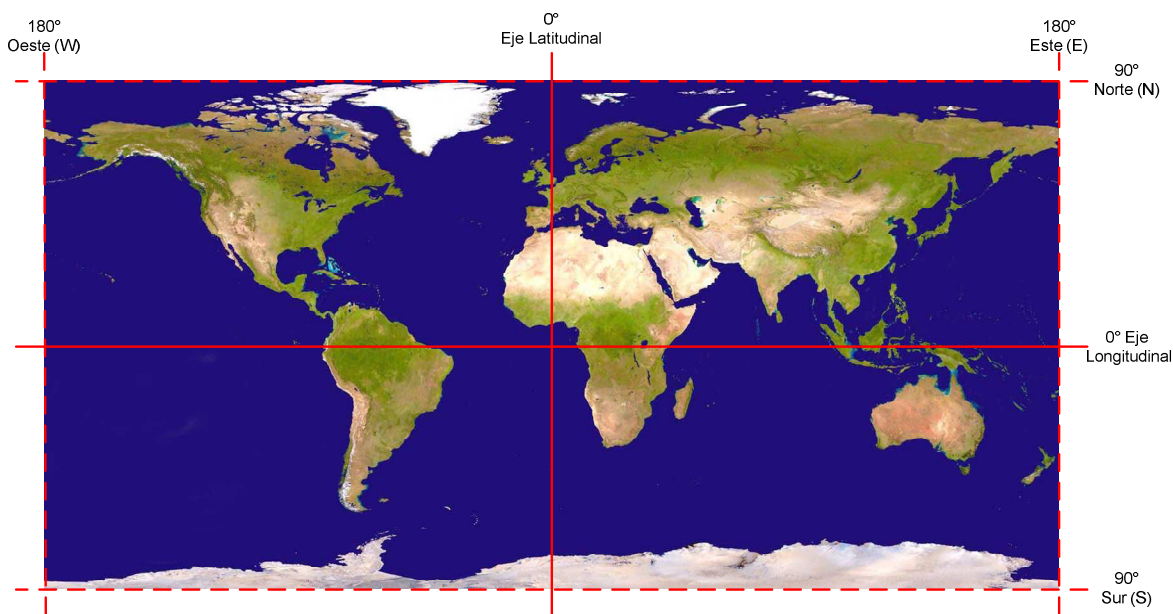
Debido a que un archivo define una y solo un área geográfica; no existirá por lo tanto ningún archivo duplicado ó repetido. El nombre de archivo tendrá correspondencia con una de las coordenadas iniciales del área de cobertura que define dicho archivo. Esto facilita el trabajo del controlador en la búsqueda del mismo.

3.1.3 Definición y Nombre de Archivo

La traslación ó desplazamiento del navegador a través del mapa se lo hace en dos dimensiones; esto es a través de un mapa mundi tipo planisferio, que es el resultado teórico de la proyección geográfica de la esfera terrestre sobre un plano. Los límites de traslación sobre el mapa son las coordenadas

180 grados Este, 180 grados Oeste en el eje longitudinal, y 90 grados Norte, 90 grados Sur en el eje latitudinal (ver siguiente gráfico).

Figura 3.1 Ejes coordenados del mapa de la tierra y sus límites ¹



El mapa de la tierra antes mencionado, se lo ha dividido en áreas de 1 x 1 minuto sexagesimal, que corresponden a un área de 1852 x 1852 metros. Cada una de estas áreas corresponde a un archivo de mapa vectorial GMV.

A cada archivo GMV se lo ha dividido en 16 partes o zonas formando una matriz cuadrada de 4 x 4. Cada zona tiene un área de 250 milésimas de minuto cuadrado sexagesimal; es decir que 4 zonas formarán 1000 milésimas de minuto ó 1 minuto sexagesimal en cada eje (ver figura 3.x).

En cada zona pueden existir 1000² puntos cuyas posiciones van desde 0 hasta 999 en cada eje. Cada punto tiene relación directa con un pixel en la pantalla. La precisión de un punto en el mapa digital será entonces de una milésima de minuto cuadrado.

El nombre de archivo consta de 11 caracteres más extensión, relacionados con una posición geográfica definida por latitud y longitud. En el

(1) Fuente: www.code.irontec.com

ejemplo de la siguiente figura, se observa el nombre de un archivo cuya coordenada superior izquierda, da el nombre al mismo (ver siguiente figura).

Figura 3.2 División en zonas del archivo GMV

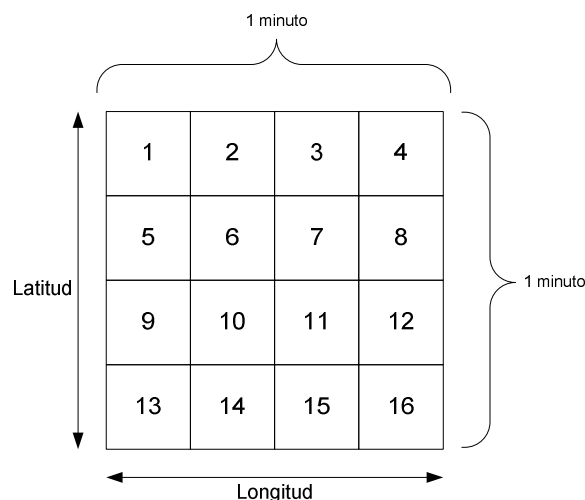
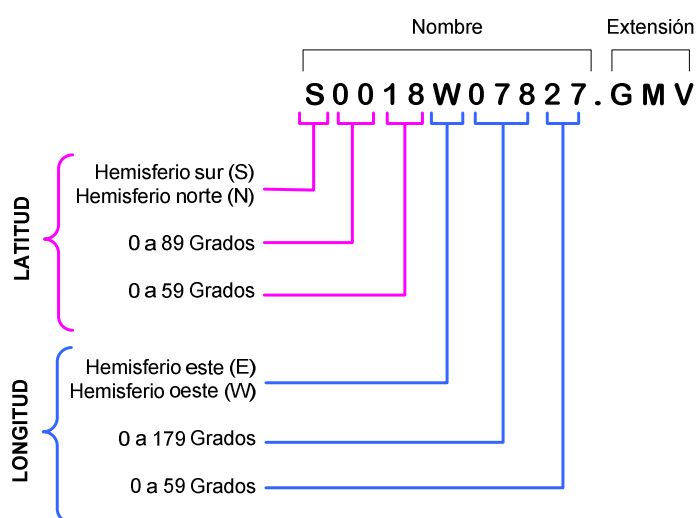


Figura 3.3 Ejemplo que explica el significado del nombre de archivo GMV



El controlador dependiendo del área geográfica de posicionamiento, solicitará los archivos que correspondan a dicha área para armar las partes del mapa dentro del sistema y dibujarlo en pantalla. El orden de posicionamiento de los archivos es uno a continuación de otro, bidimensionalmente, según las coordenadas geográficas del área en cuestión.

A continuación se observa un ejemplo de ordenamiento de cuatro archivos GMV de determinada área geográfica, sus delimitaciones y el punto que da el nombre al archivo. La separación entre archivos GMV con fines ilustrativos, no existe en la realidad; los puntos existentes entre archivos de mapa digital contiguos se dibujan uno a continuación de otro.

Figura 3.4 Ejemplo de ordenamiento de archivos GMV

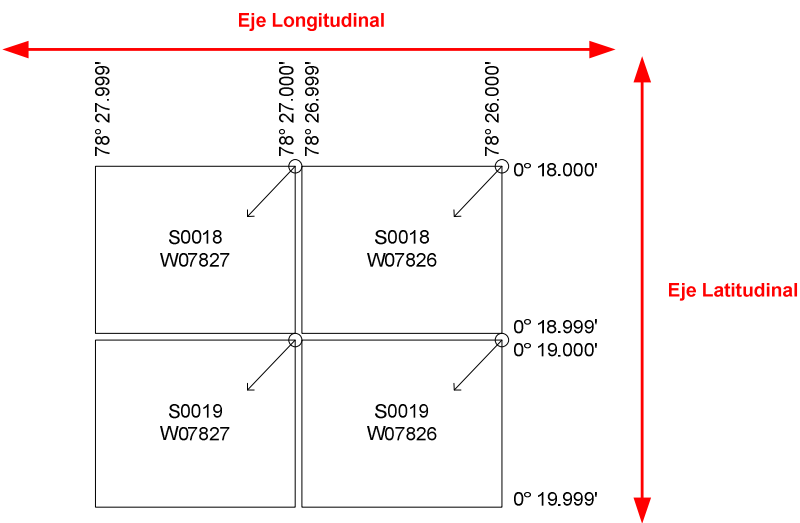
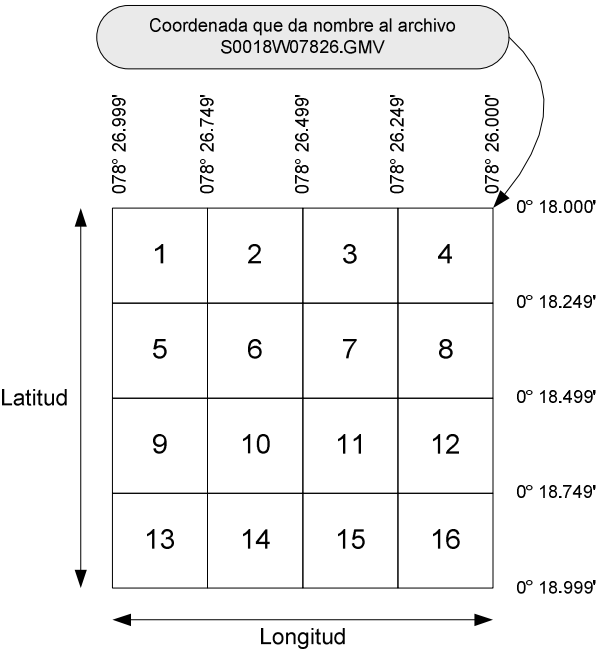


Figura 3.5 Delimitación entre zonas del archivo GMV.



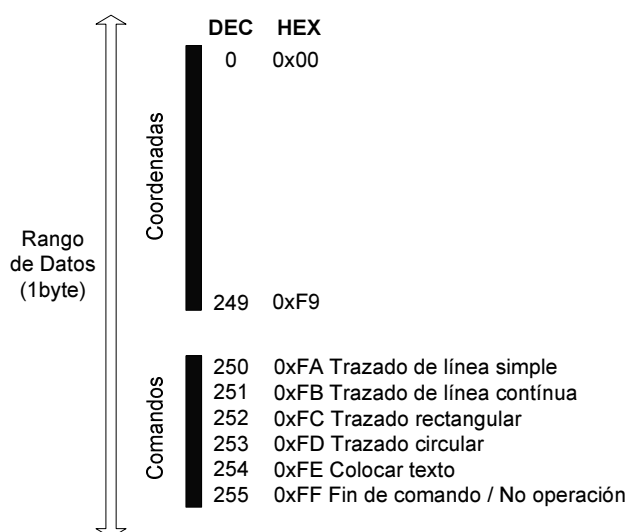
En el gráfico 3.5 se observa la delimitación de zonas de un archivo GMV en determinada área geográfica de ejemplo. Un archivo contiene 16 zonas repartidas en una matriz de 4x4. Cada zona tiene una extensión de 250 milésimas de minuto en latitud y longitud. La extensión total de un archivo GMV, como se observa, da como resultado una cobertura de 1000 milésimas en cada eje ó 1minuto latitudinal por 1 minuto longitudinal.

3.1.4 Tipo de datos

De acuerdo a lo expuesto en las consideraciones generales, los datos de navegación del archivo se forman por unidades de información de 1 byte. Esto significa que en este rango (0 a 255), se identificarán dos tipos de datos principales: Coordenadas y Comandos.

Se considera que dentro del rango de datos de 1 byte se toman los primeros 250 valores para definir una coordenada en un eje, esto es de 0 a 249. Los 6 datos restantes corresponden entonces a comandos del sistema (250 a 255).

Figura 3.6 Rango de datos que diferencian coordenadas y comandos



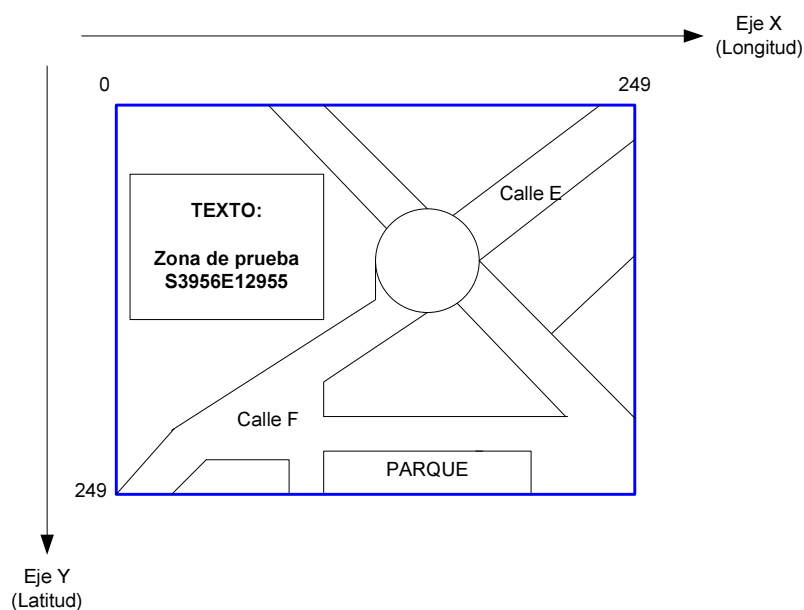
Existen seis comandos posibles: cinco operaciones gráficas definidas y una operación vacía. Estas son: trazado de línea simple, trazado de línea

continúa, trazado rectangular, trazado circular, colocar texto, y fin de comando ó no operación, como se observa en la siguiente figura.

Determinación de una “Zona”

Con un rango de datos de 250 posibilidades para definir una coordenada, queda entonces estructurada un área geográfica de 250 x 250 puntos denominada *Zona*. Una zona es una unidad de espacio virtual de trabajo para el generador de gráficas. Los comandos que definen una operación gráfica pueden trabajar dentro de esta zona únicamente.

Figura 3.7 Operación gráfica del controlador en una “Zona”



La operación gráfica mostrada en la figura anterior representa un conjunto de datos definidos sobre una "zona" construida a partir de los comandos preestablecidos, que a continuación se explican con detalle.

Definición de comandos

Las operaciones gráficas que realiza el controlador responden a seis comandos posibles: trazado de línea simple, trazado de línea continua, trazado rectangular, trazado circular, colocar texto, y fin de comando / no operación.

La estructura básica de comandos es la siguiente:

Figura 3.8 Estructura de un comando

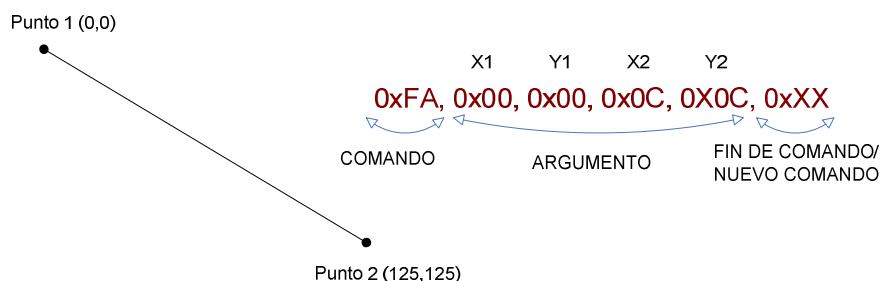


El tipo de comando se expresa con el identificador 0xFA hasta 0xFF, seguido del argumento o argumentos mas el byte de fin de comando ó también en su lugar, se puede colocar el siguiente identificador.

- **Comando 0xFA Trazado de línea simple**

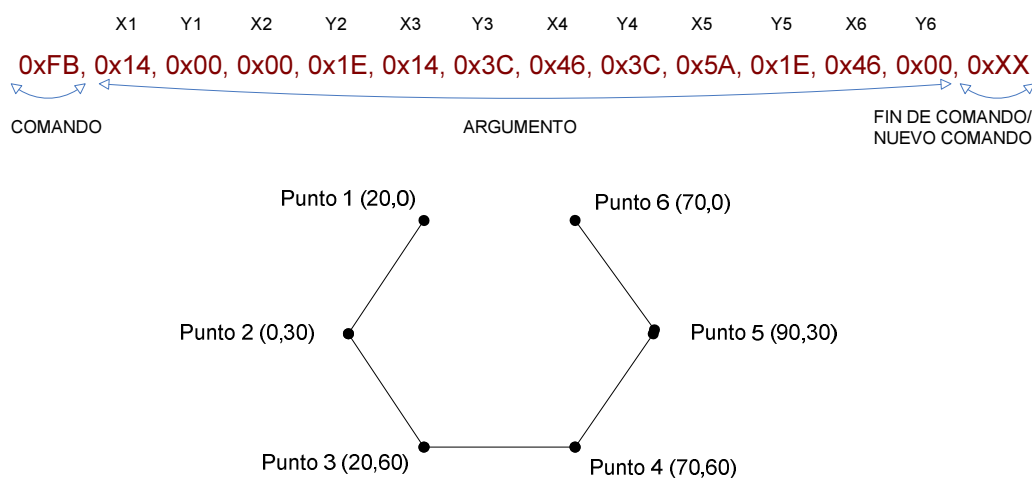
Dibuja una línea recta definida por dos puntos con la siguiente sintaxis: 0xFA, X₁, Y₁, X₂, Y₂, 0xXX. Donde el punto de inicio es (x₁,y₁) y el punto de llegada es (x₂,y₂). Ejemplo:

Figura 3.9 Ejemplo de comando 0xFA



- **Comando 0xFB Trazado de línea continua**

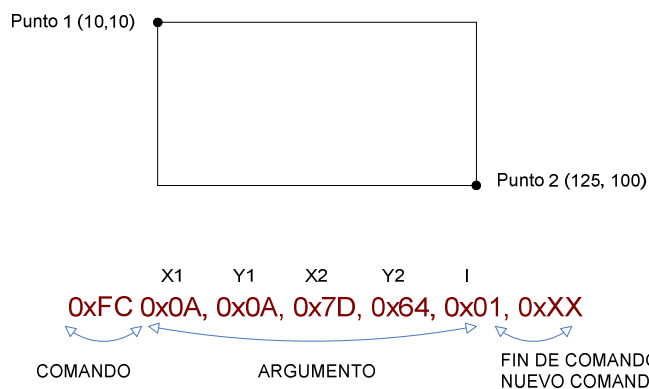
Dibuja una línea continua definida por varios puntos. Una línea continua puede partir ó no partir y llegar al mismo lugar formando un polígono. La sintaxis es la siguiente: 0xFB, X₁, Y₁, X₂, Y₂, X₃, Y₃, X_n, Y_n, 0xXX. Donde los puntos por donde pasa la línea son (x₁,y₁), (x₂,y₂), (x₃,y₃), hasta (x_n,y_n). Ejemplo:

Figura 3.10 Ejemplo de comando 0xFB

El trazado de línea continua puede o no volver al mismo punto de partida formando un polígono.

- **Comando 0xFC Trazado rectangular**

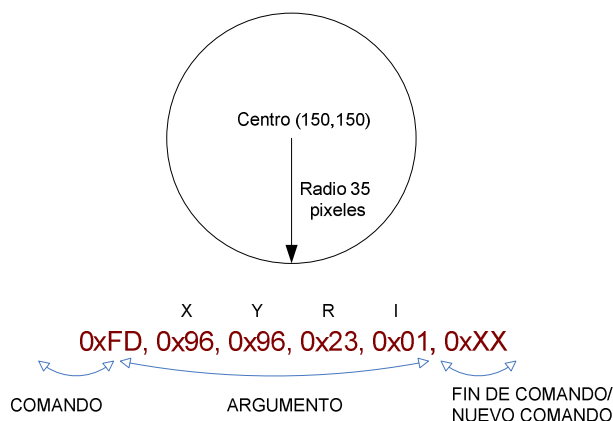
Dibuja un rectángulo definido por dos puntos. La esquina superior izquierda (X_1 y Y_1) y la esquina inferior derecha (X_2 y Y_2), y el indicador de relleno (I) que corresponde a 0: sin relleno y 1: con relleno. La sintaxis es la siguiente: `0xFC, X_1 , Y_1 , X_2 , Y_2 , I, 0xFF`. Ejemplo:

Figura 3.11 Ejemplo de comando 0xFC

- **Comando 0xFD Trazado circular**

Dibuja un círculo definido por las coordenadas del centro (X_yY), el radio en pixeles (R) y el indicador de relleno (I) que corresponde a 0: sin relleno y 1: con relleno. La sintaxis es la siguiente: 0xFD, X , Y , R , I , 0xXX. Ejemplo:

Figura 3.12 Ejemplo de comando 0xFD

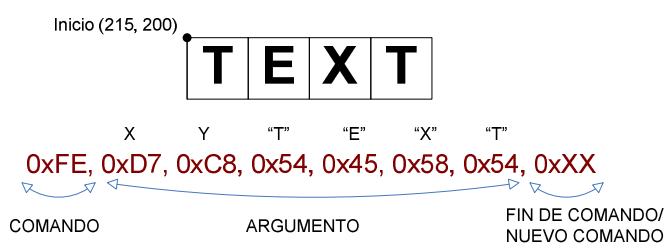


- **Comando 0xFE Colocar texto**

Dibuja un carácter ó cadena de caracteres definidos mediante la posición que se encuentra en el borde superior izquierdo del primer carácter. Después de definir la posición, a continuación se describe el carácter (C) ó cadena de caracteres en código ASCII. El número de caracteres es indefinido, y continúa hasta que el controlador encuentre un nuevo comando ó fin de comando. El tamaño de carácter es de 3x5 pixeles. La sintaxis es la siguiente:

0xFE, X , Y , ' C_1 ', ' C_2 ', ' C_3 ',, ' C_n ', 0xXX. Ejemplo:

Figura 3.13 Ejemplo de comando 0xFE



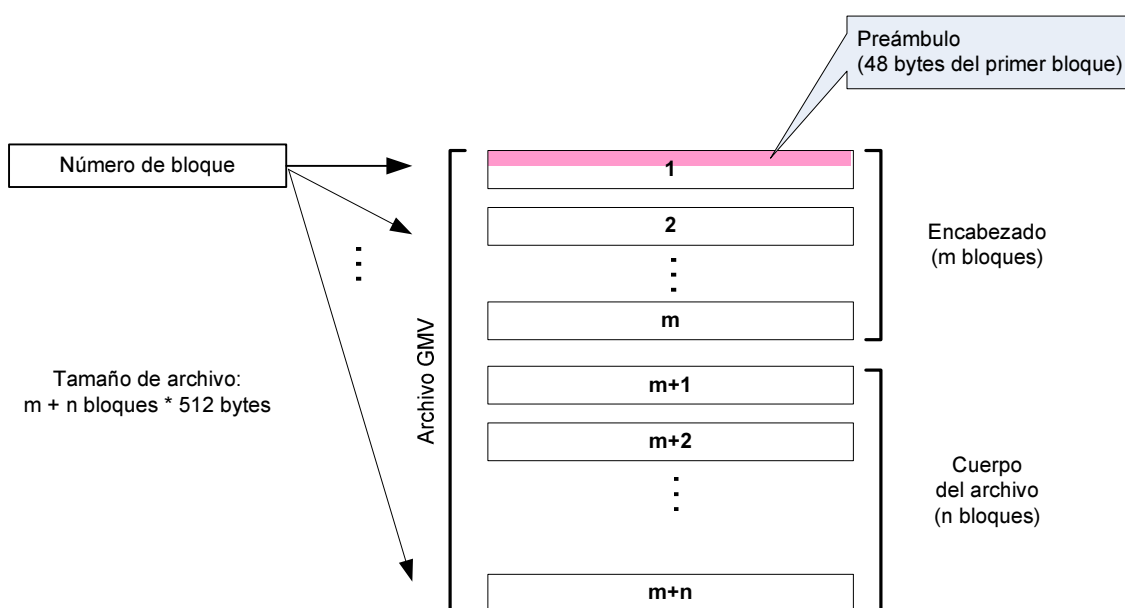
- **Comando 0xFF Fin de comando / no operación**

El comando de no operación, como su nombre lo indica permite que el controlador finalice la tarea de graficación inmediata anterior, ó también permite ignorar el dato lo que se traduce como una “no operación”.

3.1.5 Estructura interna del archivo GMV

El archivo de mapa vectorial geo-referenciado para microcontrolador (GMV), está dividido internamente en bloques de 512 bytes, donde se distinguen dos partes principales: encabezado, y cuerpo del archivo. Tanto el encabezado como el cuerpo del archivo son de longitud variable. Dependiendo de la cantidad de información que posea el archivo; el tamaño del encabezado es de m bloques y el tamaño del cuerpo del archivo es de n bloques. En el primer bloque del encabezado, existe el preámbulo del archivo con un tamaño de 48 bytes (ver siguiente figura).

Figura 3.14 Estructura interna del archivo GMV.



Para hacer referencia a un bloque del archivo en especial, se considera el número de bloque comenzando desde 1 hasta n. El preámbulo del archivo siempre se encontrará en el bloque número 1.

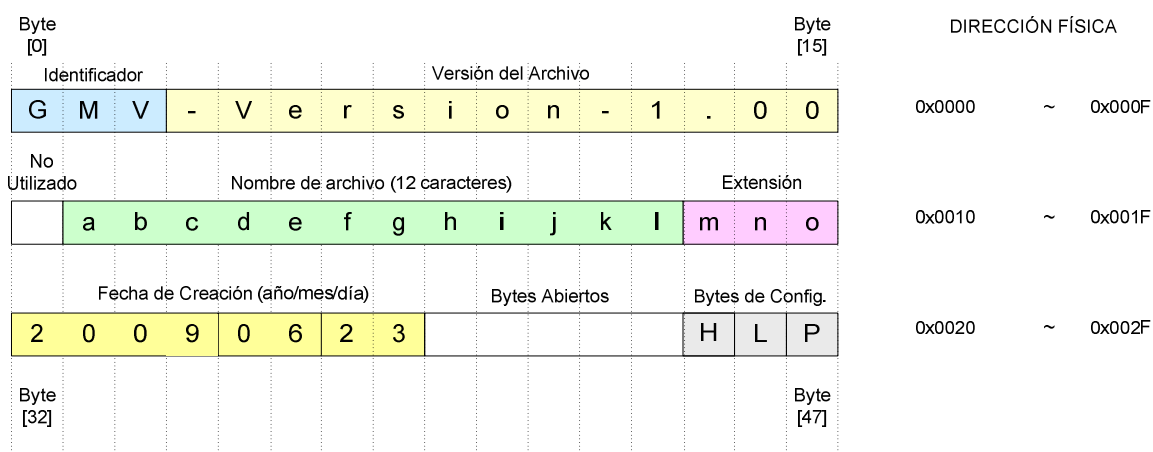
El tamaño del archivo se calculará como la suma de m bloques del encabezado más n bloques del cuerpo del archivo multiplicados por 512 bytes.

El archivo GMV está dividido en bloques de 512 bytes debido a que la unidad de lectura y escritura en la memoria SD es de tamaño de 1 bloque.

Preámbulo

En el preámbulo se encuentra la información básica necesaria para procesar el archivo por parte del sistema controlador. El preámbulo posee un tamaño fijo de 48 bytes, donde se encuentra información como: identificador de tipo de archivo, versión, nombre de archivo, extensión, área geográfica a la que corresponde, fecha de creación, tamaño de encabezado, número de capas, tamaño de puntero (ver siguiente figura).

Figura 3.15 Estructura interna del preámbulo de archivo GMV.



El controlador utilizará el identificador del archivo GMV comparando los tres primeros bytes que corresponden a los caracteres ASCII “GMV” del preámbulo del archivo, para saber que se trata de un archivo GMV, y proceder

con el resto de la lectura. La versión del archivo se describe en caracteres ASCII en los bytes 3 al 15. El byte 16 es un byte libre ó no utilizado. El nombre de archivo y extensión incluidos en este bloque (bytes 17 al 31), facilitan la tarea de alojamiento en el disco (memoria SD), y permiten nombrar el archivo de forma automática dentro del sistema del prototipo. Los bytes 32 a 39 contienen la fecha de creación del archivo en caracteres ASCII con el año, mes y día. Los bytes 40 al 44 son bytes abiertos, es decir que se pueden utilizar para alojar otro tipo de información no definida todavía.

Existen tres bytes de configuración: H, L y P. El byte H (*header*) indica al controlador, el tamaño del encabezado definido en bloques ó número 'm'. El byte L (*layer*) indica el número de capas por zona, y el byte P (*pointer*) indica el número de bytes que se utilizan para definir el tamaño de puntero de dirección ó numero de bloque.

Encabezado

En el encabezado del archivo consiste en una tabla donde se alojan los punteros de dirección. Un puntero de dirección es un byte o conjunto de bytes que contienen un número que utiliza el sistema controlador para ubicar un bloque de datos dentro del cuerpo del archivo GMV. El puntero de dirección está formado por 'p' bytes indicados en el preámbulo del archivo (*byte P - pointer*). Este tamaño de puntero va de acuerdo con la longitud en bloques del archivo; es decir que para un tamaño de puntero 'p', se podrían direccionar 2^p bloques, por lo tanto el tamaño total de archivo sería de: $2^p \times 512$ bytes.

Por ejemplo para un valor $P = 2$, quiere decir que el puntero está formado por dos bytes, es decir el puntero es un número de 16 bits, con el cual se podrían direccionar hasta un máximo de $2^{16} = 65536$ bloques, que en términos de tamaño de archivo indican un máximo de $65536 \times 512 = 33.5$ MBytes, incluyendo el encabezado y cuerpo del archivo (ver figura 3.16).

El tamaño del encabezado puede variar desde 1 bloque hasta 'm' bloques, y está definido por el byte H del preámbulo; por ejemplo con un valor

de 1 en el byte H, indica que la información de encabezado existe solamente en el primer bloque del archivo, luego del preámbulo. El encabezado se colocaría desde el byte 49 hasta el byte 511, donde culmina el bloque uno.

Figura 3.16 Ejemplo de puntero de 16 bits definido en el encabezado

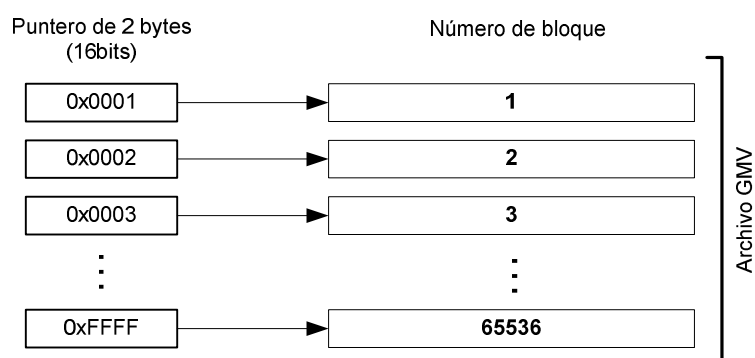
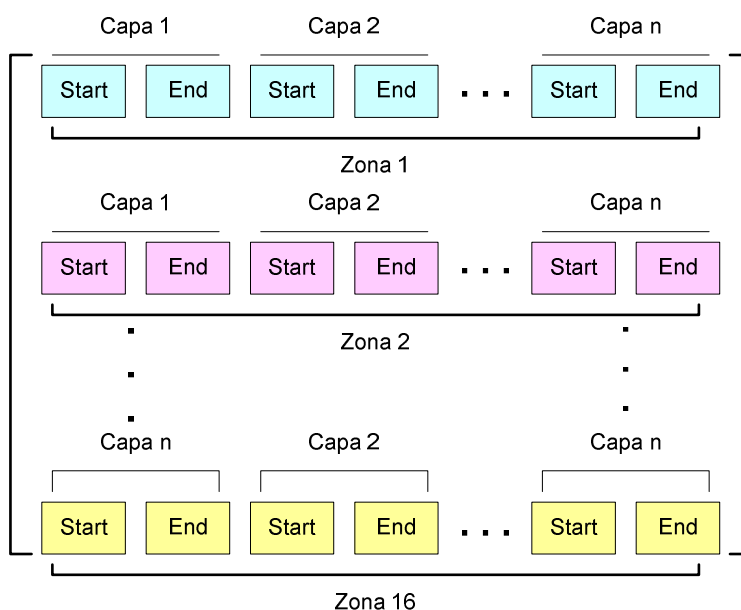


Figura 3.17 Estructura del encabezado del archivo GMV.



El tamaño del encabezado tiene concordancia con los bytes L y P, ya que el número de capas por zona, y el número de bytes por puntero, definen el tamaño de encabezado. El encabezado se encuentra inmediatamente después del último byte del preámbulo; es decir desde el byte 48. Si la especificación del

byte H es 1, entonces el preámbulo existirá solamente en el primer bloque. La estructura del encabezado consiste de punteros secuencialmente colocados que indican el rango de localización de capa en las 16 zonas existentes en el mapa. Esto se aprecia en la figura 3.17.

El puntero *Start* y el puntero *End* definen el rango de bloques donde se encuentra la información de cada capa. Esta configuración de punteros facilita el trabajo del controlador al liberarse de carga de procesamiento para que la lectura de archivos sea rápida.

La lectura del encabezado por parte del controlador, es secuencial; es decir que desde el byte de inicio del encabezado, el microcontrolador lee byte por byte hasta terminar con la última capa de la última zona. Luego de esta operación, el resto de bytes que se encuentran hasta el final del bloque, son ignorados.

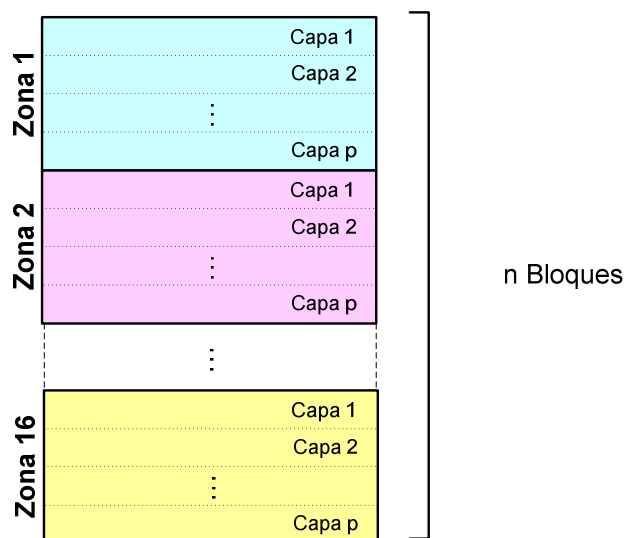
Para determinar el tamaño de encabezado, se deben tomar en cuenta el número de capas, el número de bytes por puntero, las 16 zonas del archivo, y 2 punteros para definir una capa. Para un ejemplo de 3 capas por zona y 2 bytes por puntero tenemos:

$$16 \text{ zonas} \times \frac{3 \text{ capas}}{\text{zona}} \times \frac{2 \text{ byte}}{\text{puntero}} \times 2 \text{ punteros} = 192 \text{ bytes} , \text{ lo que significa}$$

una longitud de 1 bloque ó 512 bytes, ya que no existe un tamaño menor a éste.

Cuerpo del archivo

En el cuerpo del archivo se encuentran los datos del mapa digital propiamente dicho, dividido en 16 zonas y “n” capas por zona. Todos estos datos están secuencialmente colocados.

Figura 3.18 Estructura del cuerpo del archivo GMV

El tamaño del cuerpo del archivo depende de la configuración del encabezado; es decir del número de capas, y sobre todo depende de la cantidad de información y complejidad del mapa. La cantidad de información es teóricamente infinita, pero está limitada al mismo tiempo por las características y prestaciones del sistema donde se lo lea; es decir que a pesar de no tener restricciones en el tamaño de archivo, en un sistema basado en microcontrolador, se tiene que especificar un límite de datos para desempeño óptimo del prototipo.

3.2 DISEÑO DEL NUEVO SISTEMA DE ARCHIVOS PARA MEMORIA SECURE DIGITAL (SD)

3.2.1 Introducción

Debido al volumen de datos que se requiere manejar en un disco (memoria SD), la gestión de archivos y manipulación de la información resulta algo compleja para el nivel de procesamiento de un microcontrolador; en comparación con un computador personal. En un inicio se contemplaba la posibilidad de hacer compatible la lectura de datos desde el microcontrolador

con una PC, por medio del sistema de archivos FAT (*File Allocation Table*); sin embargo la información sobre la especificación del sistema de archivos sobre memorias SD, se encontraba limitada y resultaba muy extensa en cuanto a implementación y costos de memoria de programa; entonces se tomó en cuenta la idea de crear un nuevo sistema de archivos para ser utilizado sobre la plataforma de capa física SD, y que sea gestionado por el microcontrolador.

El concepto que se planteó fue el crear un sistema de archivos sencillo que maneje volúmenes de datos extensos utilizando un nivel de procesamiento aceptable dentro de las capacidades de un microcontrolador. Tomando la estructura básica la FAT, se crea el sistema de archivos FDZ (*File Descriptor Zone*) ó Zona de Descriptor de Archivos, que propone en términos sencillos el manejo de la información digital en una memoria SD, aprovechando ciertas características del ordenamiento de datos en capa física, en base al agrupamiento de bloques de información asociados con una tabla que resume el tamaño, las características y la posición de los datos dentro del disco (memoria) para su manipulación.

3.2.2 Diseño del sistema de archivos

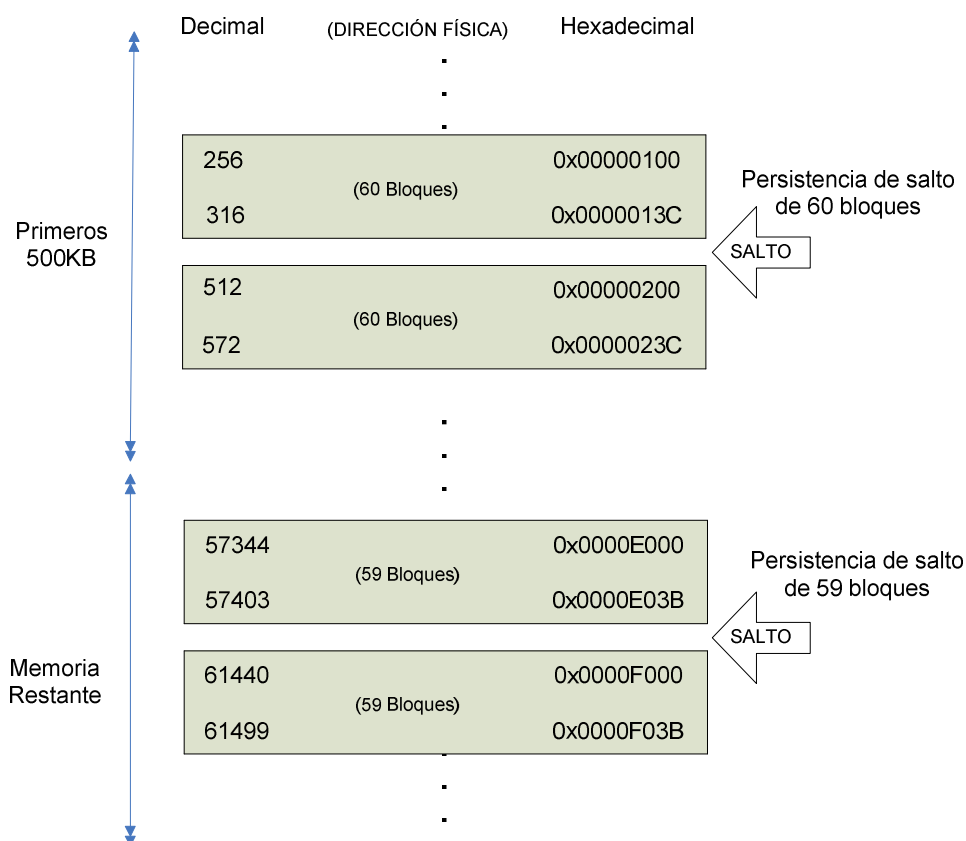
Consideraciones preliminares de la capa física

La organización de la información digital en el disco (memoria SD) se basa en bloques de datos de 512 bytes, a los cuales se acceden por medio de una dirección física hexadecimal de 8 dígitos (32 bytes), por ejemplo: 0x00FEFF3B.

Cualquier dirección de memoria no es posible, está depende de las características del disco y la capacidad de datos. La obtención de las direcciones de memoria válidas y su rango, se realizó realizando experimentación sobre una memoria SD de 1GB de capacidad. Se encontró que en los primeros 500KB aproximadamente existe un salto en el direccionamiento por cada 60 bloques de datos, es decir, el direccionamiento

no es continuo. Mientras que luego de estos primeros 500KB la persistencia en el salto ocurre cada 59 bloques. Esto se observa a en la siguiente figura.

Figura 3.19 Persistencia de salto en direccionamiento del disco SD



Esta característica intrínseca de la memoria SD de 1GB es aprovechada para diseñar el sistema de archivos FDZ planteado. La compatibilidad del sistema FDZ con el sistema FAT no será posible, debido a que el nuevo sistema de archivos manipula la memoria SD a nivel de capa física.

Esta separación o salto en el direccionamiento dará la pauta para establecer una separación de áreas importante dentro de la memoria SD, como se verá más adelante en este capítulo.

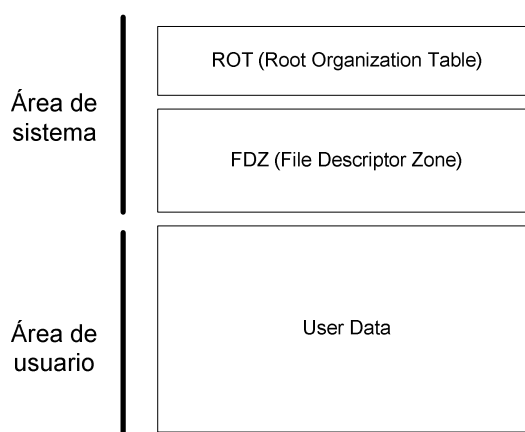
En el disco de 1GB, se logró acceder a un total de 1'765.768 bloques; es decir se podría utilizar una capacidad máxima aproximada de 900 MBytes. Debido a la aplicación de un formato al disco esta capacidad se reduce unos

pocos megabytes como se verá mas adelante. Sin embargo esta disponibilidad de tamaño de disco resulta más que suficiente para lograr los objetivos planteados en este proyecto, considerando que los archivos de prueba ocuparán tan solo unos cuantos Kilobytes.

3.2.3 Estructura general del sistema de archivos

La estructura general del sistema de archivos FDZ, se forma de tres partes ó capas cuyo orden va relacionado con la posición de memoria física dentro del disco. Se destacan también dos áreas físicas principales: el área de datos de sistema y el área de datos de usuario (ver siguiente figura).

Figura 3.20 Áreas principales del sistema de archivos FDZ



La primera y segunda capa corresponden al área del sistema. En la primera capa (ROT) de la estructura, se alojarán los datos básicos de información del sistema y el disco, tales como Identificador de sistema, capacidad de disco, nombre del volumen, número de serie, etc. En la segunda capa (FDZ) se encuentran los datos para la gestión de archivos, tales como: punteros de dirección, nombre y extensión de archivo, fecha de modificación, atributos, etc. Por último en la tercera capa (*User Data*) se encuentran los archivos de usuario en sí. Esta capa corresponde al área de usuario.

Direccionamiento

Existen dos tipos de direcciones para acceder a los datos dentro del disco. La dirección física, que está disponible para las dos áreas del sistema; y la dirección lógica, solamente presente en el área de usuario. La dirección física es el conjunto de 8 dígitos hexadecimales (32 bytes) que se utilizan para acceder a cualquier bloque de datos en todo el disco (memoria SD).

La dirección lógica es también llamada “número de unidad”, y consiste en un número decimal con el cual se hace referencia a un grupo de bloques llamados “unidades” presente solamente en el área de usuario. En la siguiente tabla se muestra la relación que existe entre la dirección lógica y la dirección física en un disco SD de 1GB.

<u>Memoria - Área de Usuario</u>			
Dirección Lógica (número de unidad)		Dirección Física (rango)	
DEC	HEX	HEX	
1	0x0001	0x00001000	0x0000103B
2	0x0002	0x00001100	0x0000113B
3	0x0003	0x00001200	0x0000123B
.	.	.	.
.	.	.	.
.	.	.	.
29912	0x70F0	0x00FEFF00	0x00FEFF3B

Tabla 3.1 Relación entre dirección lógica y dirección física

Unidades

Las unidades son grupos de 59 bloques de datos que tienen la capacidad de alojar a un archivo ó fragmento de archivo dentro de la memoria de usuario. Estas unidades son análogamente similares al clúster en FAT.

El tamaño efectivo de una unidad corresponde a 58 bloques, debido que un bloque es utilizado para control como se verá más adelante. El tamaño

efectivo de la unidad es entonces de aproximadamente 30KB y se calcula como:

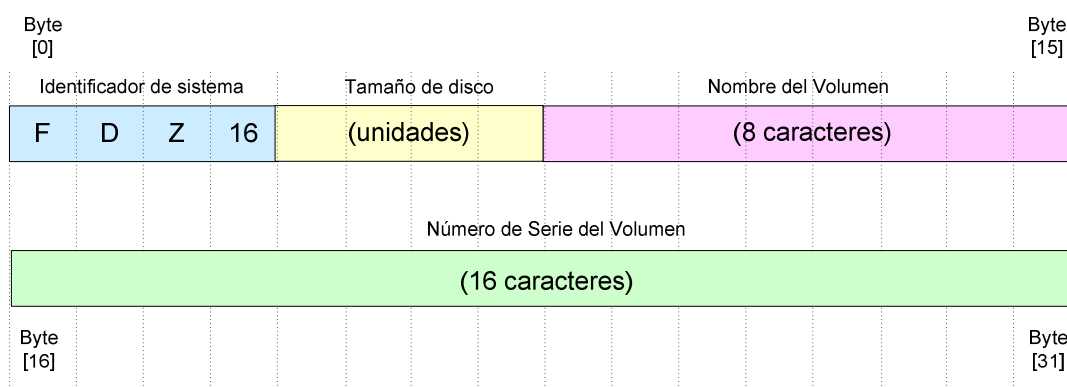
$$58 \frac{\text{bloques}}{\text{unidad}} \cdot 512 \frac{\text{bytes}}{\text{bloque}} = 29696 \text{ bytes}$$

Tabla de Organización de Raíz (ROT)

La tabla de organización de raíz (ROT – *Root Organization Table*), se encuentra en el primer bloque del disco, en la dirección física 0x00000000. Análogamente a la MBR (*Master Boot Record*) en el sistema de archivos FAT, la tabla ROT contiene la información básica del disco, la cual utiliza el controlador para identificarlo.

La tabla ROT utiliza los primeros 32 bytes del primer bloque del disco conteniendo información como: identificador de sistema, capacidad de disco, nombre del volumen, número de serie del volumen.

Figura 3.21 Estructura de la tabla ROT

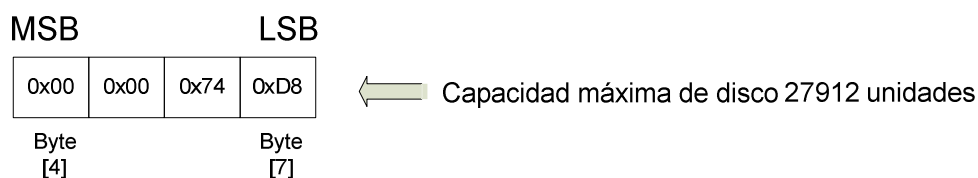


En los primeros 4 bytes de la tabla se encuentra el identificador de sistema, el cual consiste en tres caracteres ASCII “FDZ” más el número binario “16”; que el controlador reconocerá para determinar el formato del disco. Cuando el volumen no posea este identificador en su dirección 0x0000 0000, se dirá que el disco no tiene formato y un procedimiento de formateo será necesario.

Los caracteres “FDZ” son las siglas del sistema de archivos *File Descriptor Zone*, y el número 16 indica el tamaño en bits del puntero de dirección lógica utilizado para gestionar la información de usuario.

Los bytes 4 al 7 contienen la información de tamaño de disco expresado en unidades. Para un sistema FDZ16 donde el puntero de dirección es de 2 bytes, se podría expresar un número máximo de 0xFFFF ó 65536 unidades. Para el presente proyecto, utilizando un disco de 1GB de capacidad, el tamaño resulta ser de 29912 unidades ó 0x74D8 (ver figura 3.22).

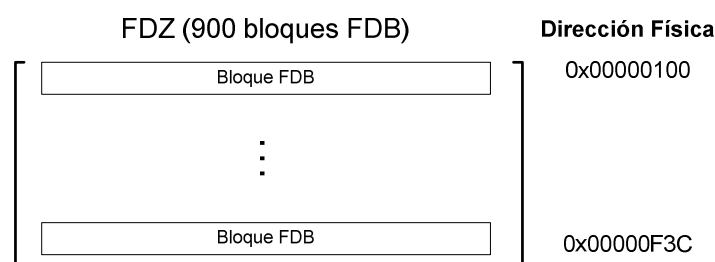
Figura 3.22 Ejemplo de indicador de tamaño de disco



Zona de Descriptor de Archivos (FDZ)

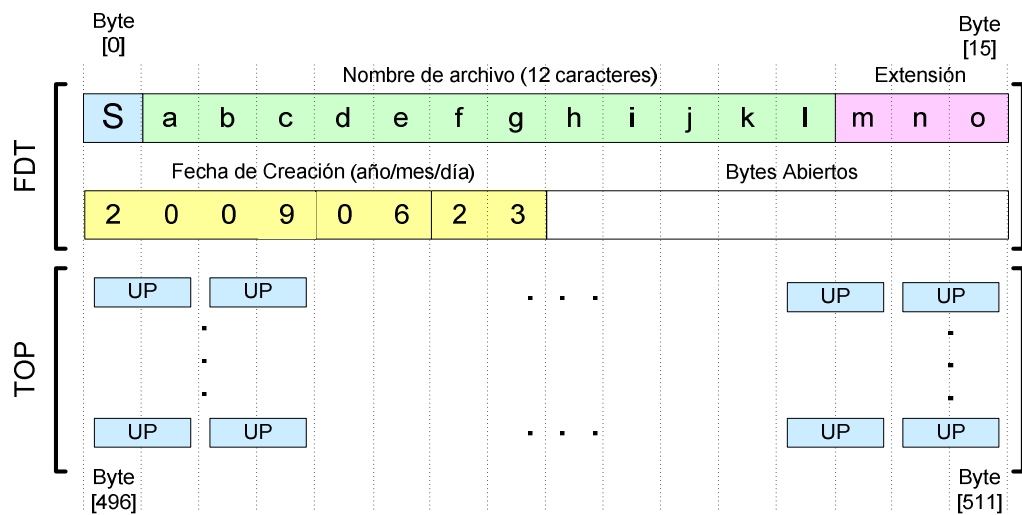
La segunda capa de la estructura del sistema de archivos es la que da nombre al sistema y se denomina FDZ (*File Descriptor Zone*) ó Zona de Descriptor de Archivos, donde se encuentran los datos para la gestión de archivos. Está formado por 900 bloques de 512 bytes denominados bloques FDB (*File Descriptor Block*) ó bloque descriptor de archivo, que contienen información para alojar hasta 900 archivos en el disco (ver siguiente figura).

Figura 3.23 Estructura de la zona de descriptor de archivos FDZ



Un bloque FDB se compone de dos partes: una tabla de datos FDT (*File Descriptor Table*) ó tabla de descriptor de archivo de 32 bytes, y una tabla TOP (*Table of unit Pointers*) ó tabla de punteros de unidad de 480 bytes (ver siguiente figura).

Figura 3.24 Estructura de un bloque FDB



Estructura de la tabla FDT

La tabla FDT corresponde a los primeros 32 bytes del bloque FDB, donde el primer byte ‘S’, indica el estado del bloque así como el tamaño del archivo en unidades. Si su valor es cero, indica que el bloque está vacío y por lo tanto el controlador puede hacer uso del mismo. Por el contrario, si su valor es distinto de cero indicará que el bloque está ocupado y además indicará el tamaño del archivo en unidades. Los valores posibles son: 0 a 240 unidades por archivo. Los bytes 1 a 15 contienen caracteres ASCII con el nombre del archivo (12 caracteres), y la extensión (3 caracteres).

El byte 16 hasta el byte 23 se utilizan para almacenar la fecha de creación del archivo, igualmente con caracteres ASCII, correspondientes a año, mes, y día.

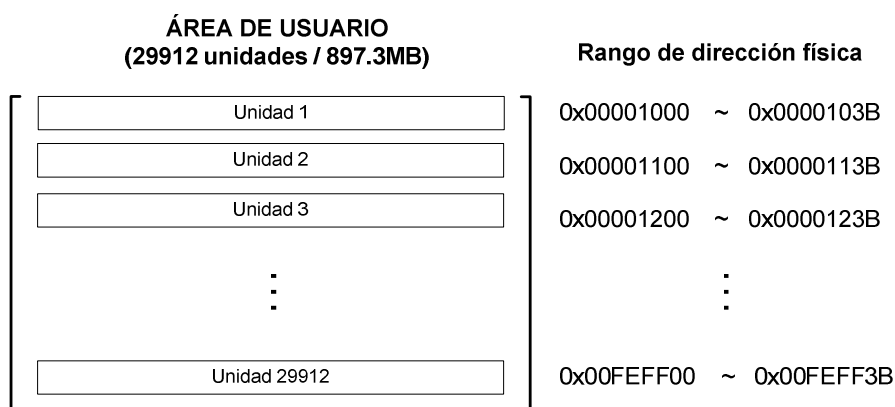
Los siguientes 8 bytes son bytes abiertos para cualquier otro tipo de información no especificada en este proyecto; como por ejemplo atributos de archivo, asociación de archivos con carpetas, etc.

Estructura de la tabla TOP

La tabla de punteros de unidad (*TOP – Table of unit Pointers*) corresponde a los últimos 480 bytes del bloque FDB, donde cada puntero de unidad (*UP – Unit Pointer*) está formado por dos bytes. Un puntero de unidad UP contiene la dirección lógica ó número de unidad donde se encuentra un archivo ó fragmento de archivo dentro del disco. Un bloque FDB en su tabla TOP puede contener hasta un máximo de 240 punteros a unidad (480 bytes / 2 bytes); lo que significa que un archivo dentro del sistema puede estar alojado en 240 unidades máximo dentro del disco, lo que implica un tamaño máximo de archivo de $240 \times 30\text{KB} = 7,2\text{MB}$.

Área de Datos de Usuario (*User Data*)

Figura 3.25 Estructura interna del área de datos de usuario



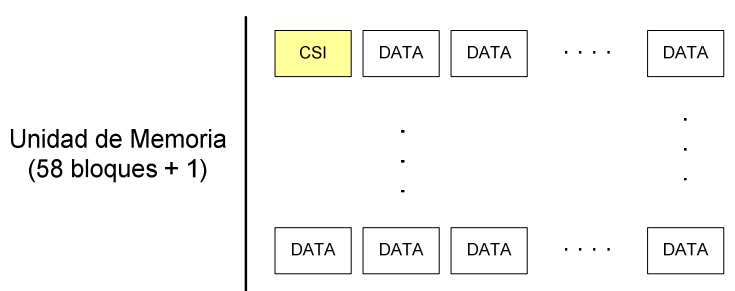
La capa de datos de usuario se define como la porción de memoria que se utiliza para guardar los archivos de usuario en el disco. El área de usuario está dividida en unidades ó grupos de $58 + 1$ bloques de 512 bytes (ver figura 3.25). El acceso a cada unidad se lo realiza por medio de la dirección lógica ó

número de unidad; que en el presente caso, por medio de un disco de 1GB de capacidad, va de 1 a 29912. Para acceder a cada bloque individualmente se utiliza la dirección física. El tamaño de área de datos de usuario se calcula como: 29912 unidades x 30KB por unidad = 897,3MB.

Dependiendo del tamaño, cada archivo se fragmenta, y cada fragmento se almacena en un espacio de memoria llamada “unidad”. La posición de la unidad dentro del disco está determinada por la dirección lógica ó número de unidad. Cada unidad está compuesta por 58 + 1 bloques de 512 bytes. La asignación del número de unidad o dirección lógica va desde 1 hasta 29912.

Cada unidad se dice que está compuesta de 58 + 1 bloques debido a que el primer bloque no es utilizado para datos de usuario; sino para información de estado de unidad. Este bloque se lo denomina CSI (*Content Status Indicator*) ó indicador de estado de contenido. La capacidad efectiva de almacenamiento de datos de usuario es entonces de 58 bloques = 29696 bytes (pese a esta corrección, se sigue considerando que el tamaño de unidad es 30KB).

Figura 3.26 Bloque CSI y unidad de memoria



El bloque CSI indica en su primer byte, el estado de la unidad; por ejemplo un valor de cero en este byte quiere decir que la unidad está libre en cuyo caso el controlador puede ocuparla para alojar un archivo ó fragmento de archivo. Cuando la unidad ha sido ocupada, el primer byte del bloque CSI toma un valor distinto de cero.

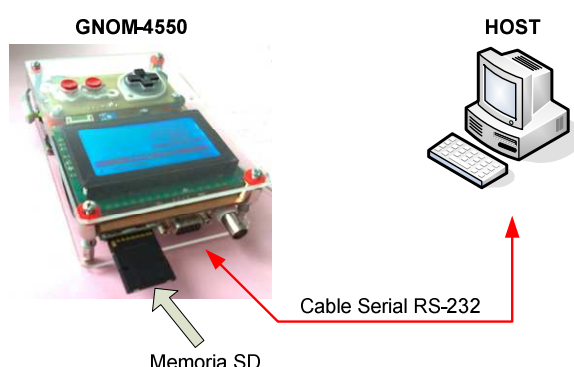
Cuando un archivo se almacena en varias unidades, la dirección de estas unidades puede ser consecutiva o no, dependiendo del estado del disco. El alojamiento de los fragmentos en el disco con dirección no consecutiva implica un fenómeno de fragmentación de la información. En el caso de las memorias SD, por ser una memoria de estado sólido, el acceso a cualquier región es de latencia similar, por lo que un fenómeno de esta naturaleza no afectaría la eficiencia del disco.

3.3 GESTIÓN DEL SISTEMA

3.3.1 Comunicación con el host

El prototipo GNOM-4550, tiene la capacidad de gestionar el sistema de archivos FDZ para un disco SD por medio del socket de memoria integrado en el hardware. El controlador tiene también la capacidad de comunicarse con un host (computador personal), por medio del puerto serial; y recibir archivos utilizando el protocolo Xmodem por medio de una aplicación como el *Hyperterminal* de *Windows*.

Figura 3.27 Comunicación con Host por medio de puerto serial

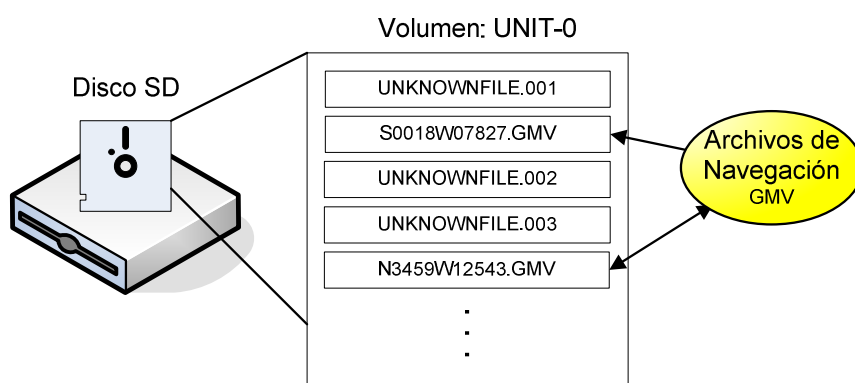


Debido a que por medio del protocolo Xmodem no se transmite información acerca del nombre del archivo que se está transfiriendo, la unidad terminal (GNOM-4550) tiene que nombrar el archivo para alojarlo dentro del disco. El controlador en este caso analiza el primer bloque del archivo recibido para saber si se trata de un archivo de navegación GMV, o cualquier otro. En el

caso de recibir un archivo GMV, el controlador utiliza la información del preámbulo del archivo para nombrarlo automáticamente en el sistema del disco; mientras que si se trata de cualquier otro archivo, el controlador le asigna automáticamente el nombre de “unknownfile.00x”.

La extensión que se asigna automáticamente es un número de 3 dígitos, que incrementa su cuenta ascendentemente con cada archivo desconocido que se almacena en el disco. De esta manera los siguientes archivos recibidos serán nombrados como: “unknownfile.002”, “unknownfile.003”, etc., (ver siguiente figura).

Figura 3.28 Identificación de archivos GMV en el sistema



La configuración del puerto serial es de: 4800bps, 8 bits de datos, 1 bit de parada, sin paridad, sin control de flujo.

3.3.2 Formateo de disco

El procedimiento de dar formato al disco consiste en escribir los datos necesarios para que el sistema lo reconozca como unidad válida dentro del prototipo GNOM-4550.

Existen dos tipos de formato de disco: alto nivel y bajo nivel. El formateo de alto nivel prepara el área de sistema del disco, mientras que el formateo de bajo nivel prepara el área de sistema y el área de usuario.

En cualquiera de los dos procedimientos de formateo de disco; el área de sistema sufre las siguientes modificaciones:

- 1) El controlador escribe la tabla ROT en la dirección 0x00000000, que contiene el identificador del sistema, el tamaño del disco, el nombre del volumen, y opcionalmente un número de serie del volumen.
- 2) El controlador borra el contenido de los 900 bloques FDB del área del sistema, que indican la existencia de archivos en el sistema.

Solamente en el formateo de bajo nivel, el controlador borra el contenido de los bloques CSI de cada unidad en el área de usuario, para asegurar que todas las unidades del disco estén libres. Debido a que el controlador borra 29912 bloques en el disco, esta operación tarda mucho más que el formateo de alto nivel; donde se procesan solamente 901 bloques.

En el caso de formateo de alto nivel, es un procedimiento más ágil que trabaja con las unidades libres del área de usuario. El disco funcionará correctamente hasta que todas las unidades estén llenas; en cuyo caso el controlador reportará un error del sistema en pantalla, y un formateo de bajo nivel será requerido para liberar todas las unidades del disco.

3.3.3 Almacenamiento de un archivo

Para realizar este procedimiento el controlador realiza una búsqueda secuencial de unidades libres dentro del área de usuario del disco; analizando el primer byte del bloque CSI de cada unidad en orden ascendente (1 a 29912). El controlador entonces coloca los fragmentos de archivo en cuantas unidades sea necesario, dependiendo del tamaño del mismo. El controlador debe llevar el registro de las direcciones lógicas ocupadas.

Cuando se ha finalizado el almacenamiento en la última unidad, el control de sistema busca un bloque vacío en el área de sistema FDZ, y coloca la

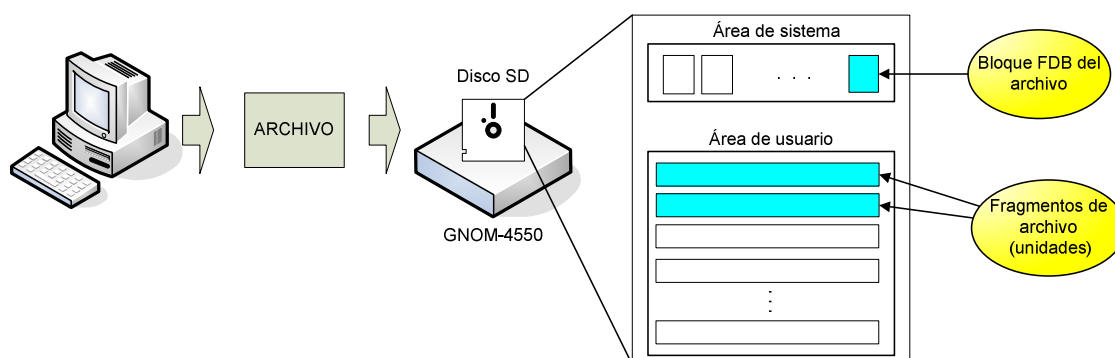
información en un bloque FDB, para registrar la existencia del archivo en el disco. Los información del bloque FDB se la realiza en dos partes: construcción de tabla FDT y construcción de tabla TOP.

La construcción de la tabla FDT se realiza de dos maneras. La primera es una construcción total cuando el archivo del que proviene es desconocido. La segunda forma es una construcción parcial cuando el archivo del que proviene es un archivo de navegación GMV.

La construcción total de la tabla FDT para alojar un archivo desconocido consiste en auto nombrar en el sistema, al archivo desconocido como “unknownfile.ext”, donde el valor de la extensión es un numero decimal de 0 a 999 dependiendo del número de archivos desconocidos se encuentren en el disco. Los bytes correspondientes a nombre y extensión se llenan con caracteres ASCII del archivo “unknownfile.ext”. Los bytes de fecha pueden ser llenados opcionalmente con la fecha del sistema. El controlador cuenta el número de unidades que ocuparon los fragmentos de archivo y escribe el byte de tamaño de archivo.

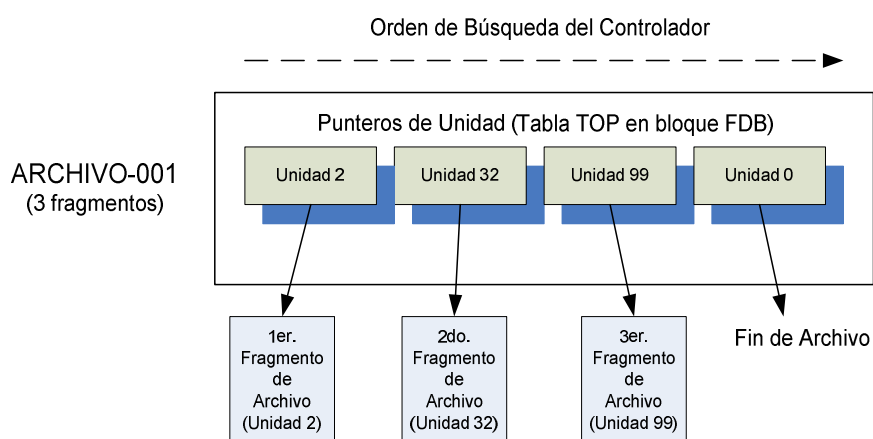
La construcción parcial de la tabla FDT para alojar un archivo de navegación GMV, consiste en copiar los bytes 16 al 47 del preámbulo del archivo GMV hacia los primeros bytes de la tabla FDT. De esta manera las características del archivo como: nombre, extensión, fecha de creación, quedan registrados automáticamente en el sistema del disco. Estos bytes se diseñaron compatibles intencionalmente para poder realizar este proceso que facilita el reconocimiento de archivos GMV dentro del prototipo GNOM-4550.

En los dos casos anteriores, la construcción de la tabla TOP es total, y consiste en escribir los bytes de punteros de unidad (UP) de acuerdo al registro de las unidades ocupadas por el archivo almacenado. Cuando todos los punteros han sido escritos, el controlador coloca al final de la tabla el número cero, que indica al sistema el fin de la cadena de punteros de unidad.

Figura 3.29 Almacenamiento de un archivo en el disco

3.3.4 Lectura de un archivo

El procedimiento de lectura de un archivo consiste en identificar primeramente el nombre y la extensión del mismo realizando una búsqueda en cada bloque FDB ocupado de la capa FDZ del área del sistema, comparando los bytes 1 al 15 de la tabla FDT. Una vez ubicado el bloque FDB correspondiente, el controlador debe procesar la tabla TOP del bloque para obtener los punteros de unidad donde se encuentran los fragmentos de archivo (ver siguiente gráfico).

Figura 3.30 Lectura de un archivo en el disco

3.3.5 Eliminación de un archivo

Para eliminar un archivo, el controlador debe ubicar el bloque FDB correspondiente en la capa FDZ. En este punto, se puede tomar dos caminos. El primero es la eliminación del bloque FDB, sin liberar las unidades ocupadas el área de usuario. Este procedimiento agiliza la tarea de borrado pero no libera recursos de memoria. Las unidades ocupadas no volverán a utilizarse aún cuando el archivo al que hayan pertenecido no exista. Este fenómeno se conoce como “cadenas perdidas”, y es similar al que ocurre en un disco duro cuando una operación de borrado no se completó por motivos como ausencia repentina de energía. En este caso el disco ó memoria, dependiendo del espacio libre, seguirá funcionando de manera correcta hasta haber ocupado completamente todas las unidades ó clústeres; y a la larga se requerirá de un formateo de disco a bajo nivel para liberar el espacio mal utilizado.

El segundo procedimiento es la eliminación del bloque FDB, además de la liberación de las unidades ocupadas, lo que implica un borrado total de la información en el medio físico. Este procedimiento asegura la no existencia de cadenas perdidas en el disco; pero requiere de una carga de procesamiento y tiempo de ejecución mayores. La liberación de unidades consiste en borrar el indicador del primer byte del bloque CSI de cada unidad ocupada.

3.4 RESUMEN DE CARACTERÍSTICAS

Las características que se resumen a continuación, pertenecen al sistema de archivos FDZ desarrollado, para un disco (memoria SD) de 1GB.

Tamaño de disco			
	Bloques	Unidades	Bytes
Área de sistema	901	No Disponible	461,312
Área de usuario	1,764,808	29912	903,581,696
	1,734,896 (efectivos)		888,266,752 (efectivos)
Total	1,765,709	No Disponible	904,043,008

Características del sistema	
Nombre del sistema	FDZ16
Tamaño de puntero	16 bits
Nombre del volumen	8 caracteres
Número de serie del volumen	15 dígitos
Número máximo de archivos	900
Tamaño máximo de archivo	7,2MB
Nombre de archivo	12 caracteres + extensión
Capacidad máxima de direccionamiento	65536 unidades / 1,966,080 bytes

Tabla 3.2 Resumen de características del sistema de archivos FDZ

CAPÍTULO 4

DISEÑO DEL SOFTWARE DE CONTROL Y PROCESAMIENTO

- 4.1 Estructura del software del sistema**
 - 4.2 Control de sistema**
 - 4.3 Control de capa física de disco**
 - 4.4 Control de formato de disco**
 - 4.5 Control de energía**
 - 4.6 Control de transferencia de archivos**
 - 4.7 Control de navegación**
-

4.1 ESTRUCTURA DEL SOFTWARE DEL SISTEMA

El software del sistema ha sido desarrollado en lenguaje C, y está formado por 7 módulos. El programa principal y 6 librerías de control. Todos los módulos fueron desarrollados por el autor del presente proyecto, excepto el módulo de control de LCD gráfico (GLCDCTRL.C – ver figura 4.1); que es una modificación de la librería “GLCD.C” del compilador CCS propiedad de *Custom Computer Services (C) Copyright 1996-2003*. La utilización y modificación de esta librería para los usuarios del compilador CCS, es permitida, como se indica en dicho archivo.

En el siguiente gráfico se muestra la estructura del software de control clasificado por librerías ó módulos de programa. Posteriormente se explicarán con detalle las funciones que realizan dichos módulos.

Figura 4.1 Estructura de módulos que componen el software del sistema

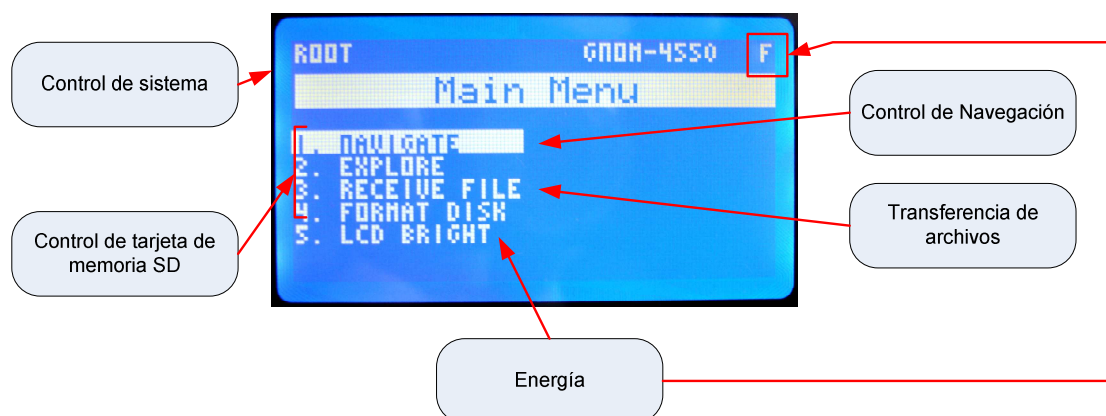


4.1.1 Descripción de grupos funcionales de programa

Para una mejor comprensión del software del sistema, se ha clasificado en grupos funcionales de programa. Un grupo funcional de programa es un conjunto de funciones que desempeñan en conjunto tareas específicas de procesamiento de determinada información ó control de una parte específica del hardware. En el prototipo GNOM-4550, existen 5 grupos funcionales que controlan los distintos elementos del sistema.

Analizando la pantalla menú principal, se logra observar a que grupo funcional se asocia cada elemento del menú.

Figura 4.2 Menú principal y grupos funcionales del sistema



El menú principal provee el acceso al usuario a 5 opciones que son:

1. Navegador GPS (*NAVIGATE*)
2. Explorador de disco (*EXPLORE*)
3. Recepción de archivo desde PC (*RECEIVE FILE*)
4. Formateo de disco (*FORMAT DISK*)
5. Control de brillo de LCD (*LCD BRIGHT*)

Los grupos funcionales que se observan en la figura anterior corresponden a:

- a. **Control de sistema.-** representado por la pantalla del menú principal y consiste en control de interrupciones, menú y submenús, ventanas de aviso, funciones gráficas menores y explorador de disco.
- b. **Control de tarjeta de memoria SD.-** como su nombre lo indica, constituyen las funciones que controlan el disco ó memoria SD y que se encuentran mayormente en la librería SDFDZ16.C; donde se pueden encontrar funciones de control de capa física y sistema de archivos FDZ.
- c. **Control de energía.-** representado por funciones de control de brillo de LCD, y medidor de nivel de batería por medio de acceso a un canal de conversión A/D.
- d. **Control de transferencia de archivos.-** representado por funciones controlan la comunicación con la PC por medio del protocolo Xmodem, escritura ó registro de información en disco SD, y creación de archivos.
- e. **Control de navegación.-** donde se encuentra el corazón del sistema conocido como núcleo de navegación, además de funciones de tratamiento de datos GPS, acceso a disco, pre-lectura, lectura, interpretación de archivos de mapa digital y graficador.

4.2 CONTROL DE SISTEMA

El grupo funcional “control de sistema” como su nombre lo indica se encarga tanto del control y la configuración básica del microcontrolador, como del sistema en general. El control del sistema se encuentra mayormente en el

archivo de programa GNOM4550.C y parte en SUBPRG.C. Las funciones más importantes que se encarga el control del sistema son:

- Configuración básica de microcontrolador
- Control de interrupciones
- Memoria RAM y registros principales
- Ventanas de aviso
- Funciones gráficas menores
- Explorador de disco

4.2.1 Configuración básica de microcontrolador

La configuración básica del microcontrolador contempla las líneas de código necesarias para que exista consistencia lógica entre el hardware implementado y las líneas del microcontrolador. Estas configuraciones son de definición y velocidad de puertos, oscilador externo, líneas de datos, entre otras. Al ser una configuración básica no se aborda con mayor detalle en este informe, sin embargo el detalle del hardware y su configuración se estudian en el capítulo 5.

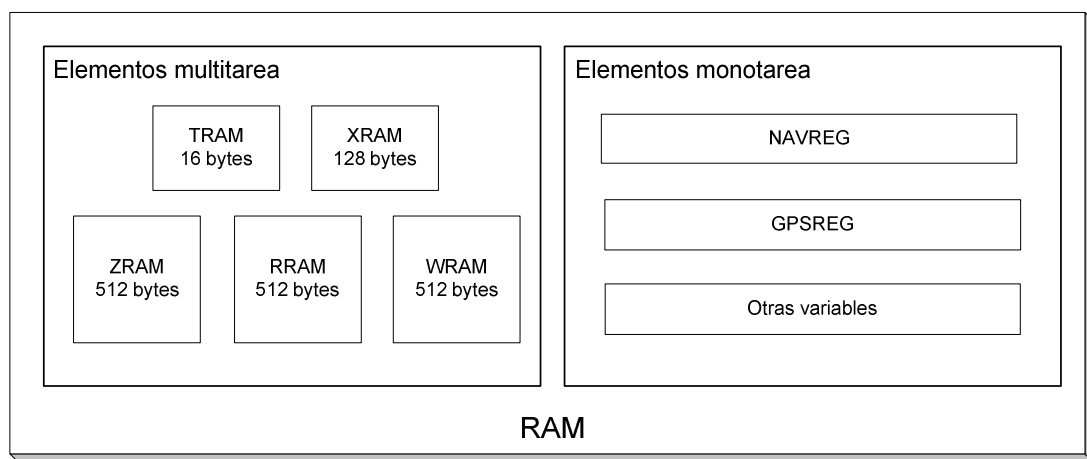
4.2.2 Control de interrupciones

El programa principal utiliza dos interrupciones de temporización. La primera es una interrupción para captura de caracteres de los mensajes NMEA0183 desde el receptor GPS. (Este tiempo de interrupción se analiza en grupo funcional “control de navegación”. La segunda interrupción es utilizada para llevar un control de tiempo en milisegundos para distintos eventos del sistema como: tiempo de espera de recepción de paquetes Xmodem, tiempo de inicialización de receptor GPS, refrescamiento de datos de navegación, entre otros.

4.2.3 Memoria RAM y registros principales

La memoria RAM del sistema forma parte de la memoria volátil interna del microcontrolador. A continuación se observa su estructura interna.

Figura 4.3 Estructura interna de la memoria RAM del sistema



La memoria RAM está dividida en dos grupos: los elementos multitarea y monotarea. Los elementos multitarea como su nombre lo indica, no se destinan únicamente a una sola función, sino que trabajan en distintas aplicaciones dentro del programa de control; y éstos son WRAM, RRAM, TRAM, ZRAM, y XRAM. Los elementos monotarea se destinan únicamente para una función determinada; y las componen dos registros principales: registro de navegación (NAVREG), registro GPS (GPSREG), y otras variables de programa.

A continuación se muestran las funciones de los elementos de memoria RAM mediante la siguiente tabla:

Elemento de memoria	Función	Grupo
XRAM	Buffer de recepción de paquetes Xmodem , buffer de tramas de datos del receptor GPS.	multitarea
RRAM	Buffer de lectura de bloque en disco SD.	multitarea
WRAM	Buffer de escritura de bloque en disco SD, almacenamiento temporal de nombres de archivo de navegación.	multitarea

ZRAM	Memoria caché de almacenamiento de tablas NFDI, construcción de bloques FDB.	multitarea
TRAM	Almacenamiento de tabla ROT, conversión de nombre de archivo de navegación.	multitarea
NAVREG	Registro de variables de navegación	monotarea
GPSREG	Registro de variables GPS	monotarea
Otras Variables	Otras variables de programa	monotarea
Total RAM aproximado:		1,82 Mbytes

Tabla 4.1 Elementos de memoria RAM y sus funciones

4.2.4 Ventanas de aviso

Las ventanas de aviso son mensajes de información del sistema que se presentan en pantalla, y que son accionados por diversas fuentes dentro del programa en ejecución. Los mensajes se clasifican en tres tipos: Alerta, información y confirmación. Por ejemplo:

- ✓ Mensajes de Alerta: Tarjeta no está lista ó no está presente
- ✓ Mensaje de Información: Formato de disco / Borrando bloques de capa FDZ
- ✓ Mensajes de confirmación: Borrar Archivo? Si / No

Figura 4.4 Ejemplo de ventana de aviso con mensaje de alerta



4.2.5 Funciones gráficas menores

Las funciones gráficas menores que se encuentran en esta parte del programa corresponden a funciones de graficación de barra de progreso, representación numérica directa desde variables enteras.

4.2.6 Explorador de disco

El explorador de disco es un subprograma que se encarga de visualizar en pantalla el contenido del disco SD, mostrando los archivos existentes por medio del nombre, extensión, tamaño de archivo, número de archivos totales, y espacio libre. Es posible también eliminar uno ó varios archivos del disco. A continuación se muestra un ejemplo de la pantalla de explorador conteniendo un archivo.

Figura 4.5 Pantalla de explorador de disco



EL explorador de disco realiza la búsqueda de todos los 900 bloques FDB del área de sistema de disco en la capa FDZ, leyendo la tabla FDT de cada bloque FDB ocupado por un archivo. En dicha tabla se encuentra el nombre de archivo mas extensión y el tamaño de archivo en unidades. Para obtener el tamaño de archivo en kilobytes, el explorador multiplica el número de unidades por 30 que es el tamaño de unidad en Kilobytes (ver cap. 3).

Número de unidades de archivo: 5

5 unidades 30 KB por unidad

Tamaño de archivo = 150KB

El cálculo de espacio libre se realiza sumando el número de unidades ocupadas por cada archivo y restándolas del total de unidades disponibles en el disco (28912). Luego se transforma el resultado a Megabytes. Ejemplo:

$$\begin{aligned} & \text{Número de archivos: 1; Número de unidades de archivo: 5} \\ & (29912 \text{ unidades disponibles} - 5 \text{ unidades}) * 0.03 \text{MB por unidad} \\ & \text{Espacio libre} = 867.21 \text{ MB} \end{aligned}$$

La eliminación de un archivo consiste en ubicar por medio del cursor el archivo que se desea eliminar, y pulsar el botón B. Es entonces cuando el sistema dispara un mensaje de confirmación de borrado (ver siguiente figura).

Figura 4.6 Mensaje de confirmación de archivo a borrar



Al confirmar el comando, el proceso de borrado ubica el número de bloque FDB de la capa FDZ en el área de sistema del disco y borra su contenido. Esto se explica con mayor detalle en el capítulo 3. Al terminar el proceso, el contenido del disco es actualizado en pantalla.

4.3 CONTROL DE CAPA FÍSICA DE DISCO

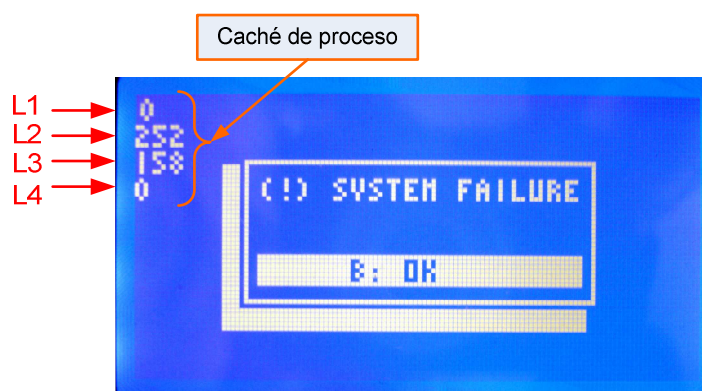
El grupo funcional “de control de capa física de disco” como su nombre lo indica realiza funciones de control a nivel de capa física de la tarjeta de memoria SD a través del bus de comunicación SPI. Se encuentra implementado sobre la librería de programa SDFDZ16.C. Las principales funciones son las siguientes:

- Secuencia de Inicialización

- Lectura bloque
- Escritura de bloque

Con el fin de llevar un control interno del comportamiento de estas funciones son respecto al hardware, el sistema posee un registro llamado “caché de proceso”, que contiene 4 niveles de almacenamiento: L1, L2, L3 y L4. Este registro guarda información de respuesta de comandos, número de ciclos de ejecución, entre otros. De esta manera cuando existe un error general de disco, en la pantalla aparecerán los datos del caché de proceso almacenados.

Figura 4.7 Pantalla de error de sistema con información de caché de proceso



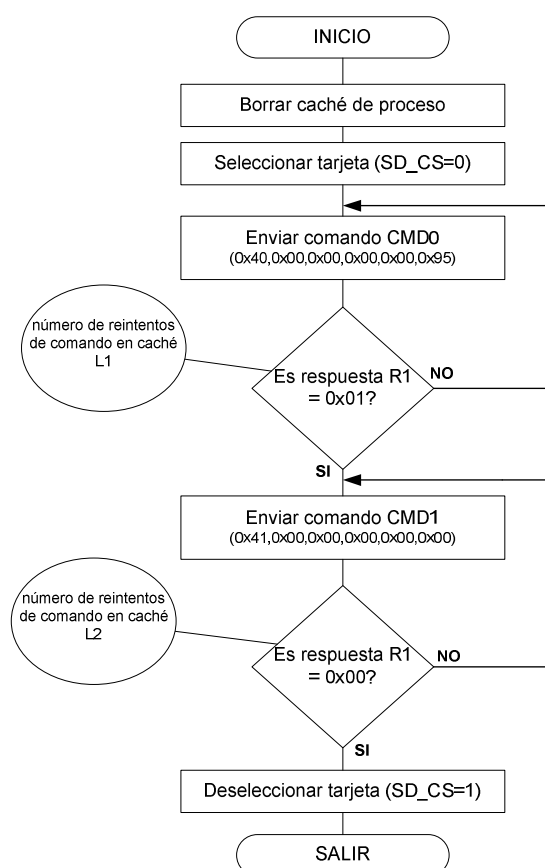
La información que contienen los distintos niveles del caché de proceso y su explicación se observan en el detalle de las siguientes funciones principales de control de capa física:

4.3.1 Secuencia de inicialización

El procedimiento de inicialización de modalidad SPI de la tarjeta de memoria se lo realiza después de conectar la alimentación ó después de haber insertado la tarjeta en el slot; debido a que la tarjeta por defecto enciende en modo SD.

El host debe entonces cumplir con los pasos especificados en el proceso de encendido para inicializar la tarjeta correctamente (ver cap. 2). Una vez superado el tiempo de rampa de encendido aplicando varios ciclos de reloj en el bus SPI, la tarjeta reconocerá la petición del host para trabajar en modalidad SPI cuando en el pin de selección SD_CS, exista un nivel lógico bajo (GND), mientras se envía el comando CMD0 (*GO_IDLE_STATE*), con su correspondiente byte CRC correcto (0x95); debido a que la tarjeta todavía no se encuentra en modalidad SPI. Es entonces que la tarjeta responderá con un byte tipo R1 al comando, indicando el estado: *IDLE*, mediante el bit correspondiente (0x01). Esta operación se la realiza de manera repetitiva hasta recibir esta respuesta. Inmediatamente, el host debe llamar constantemente al comando CMD1 (*SEND_OP_CONDITION*), hasta recibir en el bus SPI, la respuesta R1 0x00. Es entonces cuando el proceso de inicialización ha sido completado y la tarjeta SD se encuentra lista en modalidad SPI, lista para recibir el siguiente comando.

Figura 4.8 Diagrama de flujo simplificado de la función “init_sequence”



En la librería de control SDFDZ16.C, se ha construido la función: *init_sequence* para realizar esta tarea. Los procesos internos acumulados en el caché de proceso por parte de esta función son los siguientes:

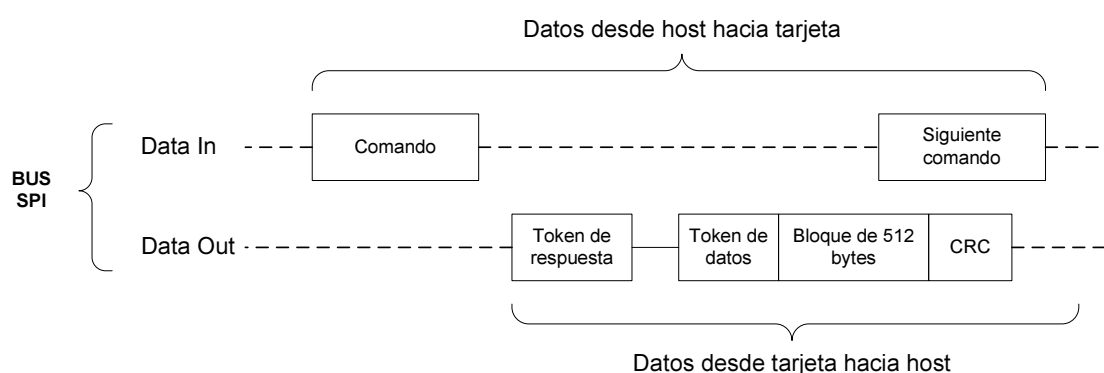
Error Caché de Proceso	L1	reintentos comando CMD0 (max 10,000)
	L2	reintentos comando CMD1 (max 10,000)
	L3	no utilizado
	L4	no utilizado

Tabla 4.2 Información en el caché de proceso para la función de inicialización

4.3.2 Lectura de bloque

Realiza el procedimiento de lectura de bloque del disco a partir de una dirección física de 4 bytes depositando la información en RRAM. En modalidad SPI, el comando enviado a la memoria SD es *READ_SINGLE_BLOCK* (CMD17). Una vez aceptado el comando la tarjeta enviará un token de respuesta seguido del bloque de datos con longitud 512 bytes. Por último la memoria envía dos bytes de CRC que en el presente caso se ignoran.

Figura 4.9 Procedimiento de lectura de bloque en disco SD



A continuación se mencionan los pasos del proceso interno para realizar una lectura de bloque en la memoria SD:

- Activar tarjeta (SD_CS=0)

- Envío de comando: CMD17 y lectura en bloque 0
(0x51,0x00,0x00,0x00,0x00,0xFF,0xFF)

ComandoDirecciónCRC (cualquier valor)
- Esperar respuesta R1 (ver cap. 2):
0x00 -> OK
0x20 -> Error de dirección (la operación se cancela)
- Esperar token de datos (ver cap. 2) de acuerdo a la operación realizada:
0xFE -> Operación Válida
0x08 -> Operación No Válida
- Recibir trama de datos del bloque de 512 bytes
- Esperar 2 bytes de CRC (ignoradas en este caso)
- Desactivar tarjeta (SD_CS=1)

Para este caso se ha desarrollado la función *read_block*, cuyo argumento es la dirección física de 4 bytes del bloque que se requiere leer, con byte MSB primero:

Ejemplo: *read_block* (0x00, 0x00, 0x00, 0x00).

Los errores y el estado del proceso interno de la función “*read_block*”, se acumulan en el caché de proceso con el siguiente contenido:

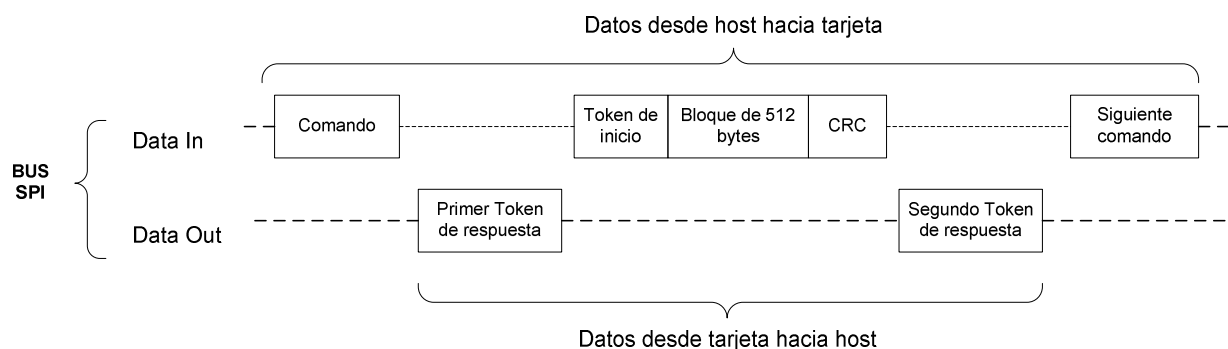
Caché de Proceso	L1	token de respuesta a comando
	L2	token de datos
	L3	reintentos hasta recibir token de datos
	L4	número de ciclos hasta vaciar el bus (max. 65,536)

Tabla 4.3 Contenido de caché de proceso para la función “*read_block*”

4.3.3 Escritura de bloque

Realiza el procedimiento de escritura de un bloque de memoria en el disco con la información de WRAM, a partir de una dirección física de 4 bytes. El comando SPI correspondiente es *WRITE_BLOCK* (CMD24), donde la tarjeta responderá con un token tipo R1 y esperará al envío de la trama de datos o bloque desde el host, encabezado por un token de inicio. Se envían finalmente 2 bytes de CRC que se ignoran. El direccionamiento es idéntico al procedimiento de lectura. La longitud de bloque válida es 512 bytes.

Figura 4.10 Procedimiento de escritura de bloque en disco SD



Para este caso se ha desarrollado la función *write_block*, cuyo argumento es la dirección física de 4 bytes del bloque en el cual se requiere escribir, con byte MSB primero:

Sintaxis: *write_block* (byte4, byte3, byte2, byte1).

Ejemplo: *write_block* (0x00, 0x00, 0x00, 0x00).

A continuación se mencionan los pasos del proceso interno de la función para realizar una escritura de bloque simple en un sector de la memoria SD:

- Activar tarjeta (SD_CS=0)
- Envío de comando: CMD24 y lectura en bloque de dirección 0

(0x58,0x00,0x00,0x00,0x00,0xFF,0xFF)

Comando **Dirección** **CRC (cualquier valor)**

- Esperar primer token de respuesta R1 (ver cap. 2):
 - 0x00 -> OK
 - 0x20 -> Error de dirección (la operación se cancela)
- Enviar byte de inicio 0xFE
- Enviar bloque de datos (512 bytes)
- Enviar 2 bytes de CRC (cualquier valor en este caso)
- Recibir segundo token de respuesta (ver cap. 2) mas dos bytes adicionales
 - 0xE5 -> Operación válida
 - 0xED -> Error de escritura
- Desactivar tarjeta (SD_CS=1)

Los errores y el estado del proceso interno de la función “*write_block*”, se acumulan en el caché de proceso con el siguiente contenido:

Caché de Proceso	L1	primer token de respuesta a comando
	L2	segundo token de respuesta a comando
	L3	reintentos hasta recibir segundo token de datos
	L4	número de ciclos hasta vaciar el bus (max. 65,536)

Tabla 4.4 Contenido de caché de proceso para la función “*write_block*”

4.4 CONTROL DE FORMATO DE DISCO

El grupo funcional “control de formato en disco SD” es el encargado de verificar el formato de la tarjeta así como el de dar formato a la misma; bajo el sistema de archivos FDZ. Se encuentra implementado sobre la librería de programa SDFDZ16.C. Las principales funciones son las siguientes:

- Chequeo de formato de disco
- Formatear disco

4.4.1 Chequeo de formato de disco

La función de chequeo de formato realiza las siguientes funciones:

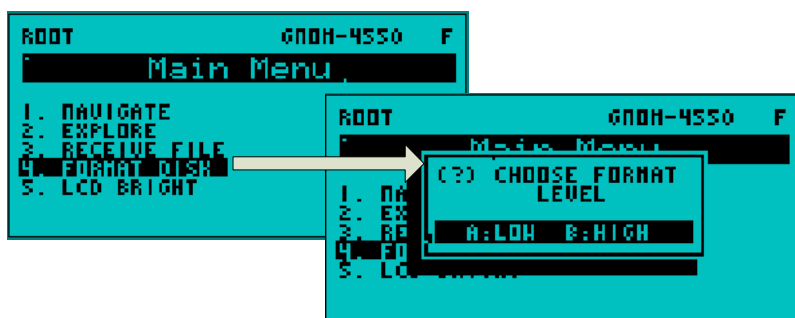
- 1) Inicialización de la tarjeta.
- 2) Lectura de tabla ROT.
- 3) Reconocimiento del identificador de sistema en la tabla ROT.

Si el identificador es “FDZ16”, entonces la tarjeta se encuentra formateada y el sistema continúa con el siguiente proceso; caso contrario un mensaje de tarjeta no formateada será desplegado. (Información detallada en el capítulo 3).

4.4.2 Formateo disco

La función de formateo de disco da formato a la tarjeta SD bajo el sistema de archivos FDZ. Existe dos tipos de formato: alto nivel y bajo nivel (explicado en el capítulo 3). En la siguiente gráfica se observa la pantalla donde se escoge el nivel de formateo deseado.

Figura 4.11 Pantalla de tipos de formateo de disco



Las operaciones internas son las siguientes:

- 1) Escribir los datos de la tabla ROT que se encuentran en la memoria ROM del procesador hacia el primer bloque de memoria en la dirección 0x00000000, con la siguiente información:

Figura 4.12 Contenido en ROM de tabla ROT

0x46, 0x44, 0x5A, 0x10, 0x00, 0x00, 0x70, 0xF0, 0x55, 0x4E, 0x49, 0x54, 0x2D, 0x30, 0x30, 0x31,							
File System ID				Max Size in units			
F	D	Z	16	28.912 units			
				U N I T - 0 0 1			

- 2) Borrar bloques de capa FDZ en el área de sistema del disco (900 en total). El rango de dirección física va desde: 0x00000100 a 0x00000F3C.
- 3) Borrar contenido de bloques CSI del área de usuario en el disco SD, con el rango de dirección lógica 1 hasta 28912. Este proceso es ejecutado únicamente en el formateo de disco de bajo nivel (ver cap. 3).

Una vez completado este conjunto de procedimientos, la tarjeta de memoria SD quedará formateada y sin dato alguno, bajo el sistema de archivos FDZ.

4.5 CONTROL DE ENERGÍA

El grupo funcional “control de energía” como su nombre lo indica lo componen las funciones que se relacionan con la energía del sistema. Este grupo funcional se encuentra mayormente implementado en el programa principal GNOM4550.C. Las funciones principales son:

- Control de brillo de pantalla LCD
- Visualizador de nivel de batería

4.5.1 Nivel de brillo de pantalla

La última opción del menú principal corresponde al control de brillo de pantalla LCD, en la cual por medio de los cursores de teclado izquierdo y derecho, se varía la intensidad luminosa de los led de luz de fondo de la pantalla. Este control se lo realiza por medio de un canal de modulación de ancho de pulso PWM del microcontrolador.

Por medio de la sentencia `set_pwm1_duty(var)`; se puede controlar el nivel de modulación del generador PWM; donde la variable “var”, es un número de 10 bits que varía entre 0 y 1023 posibilidades, y corresponde a un ancho de pulso de 0 a 100% respectivamente.

La frecuencia de modulación trabaja con el temporizador TIMER2, que también es utilizado para proveer de referencia temporal en el sistema. Como la interrupción de temporización ocurre cada milisegundo entonces la frecuencia del oscilador PWM es el inverso de este periodo; es decir 1KHz.

Figura 4.13 Pantalla de control de brillo de GLCD



4.5.2 Nivel de batería

La unidad GNOM-4550 trabaja con dos tipos de modalidad de fuente de potencial; con adaptador externo ó con batería recargable de 9V incorporada en el prototipo. En cualquiera de los dos casos existe la posibilidad de medir el nivel de carga de la batería por medio de un canal de conversión A/D

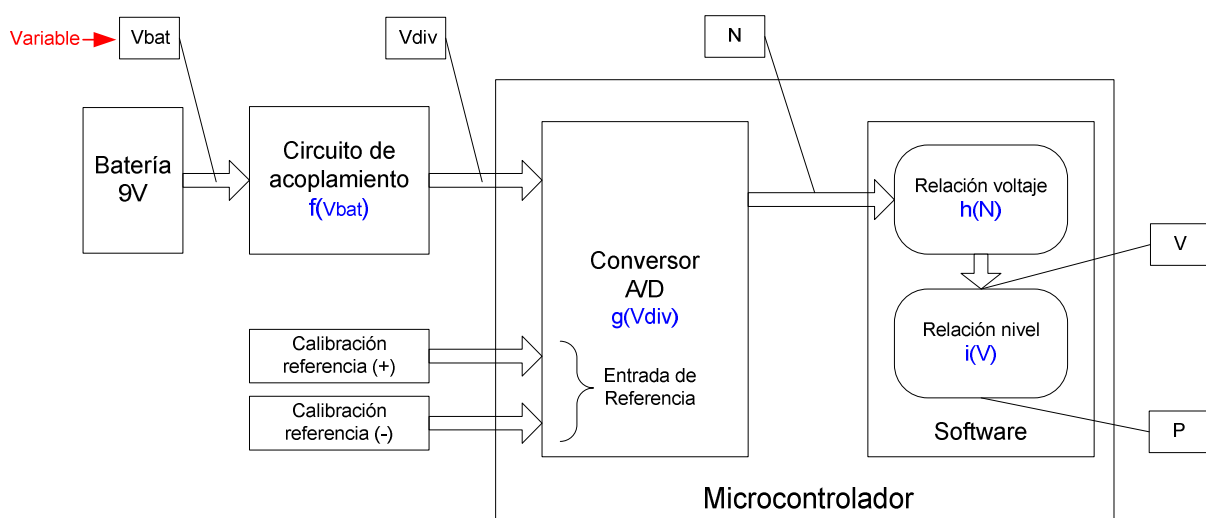
(análogo/digital) del microcontrolador. La configuración de hardware se encuentra en el capítulo 5.

El voltaje mínimo de carga de batería no debería descender de 7 voltios debido las características de regulación del circuito integrado 7805, que provee de 5V al sistema. De esta manera, teniendo en cuenta que se utiliza una batería de 9 voltios, el nivel de carga de batería sería de 100% para su valor máximo de 9V y 0% para un valor mínimo de 7V, obtenido experimentalmente y verificado en la hoja de datos del circuito integrado regulador de voltaje.

Para obtener en el dominio de software el nivel de carga de batería, se deben tener en cuenta las distintas etapas que atraviesa dicha señal, hasta llegar al conversor A/D; aplicando las funciones de transferencia correspondientes.

En la siguiente figura se observa que la señal de carga de batería V_{bat} atraviesa el circuito de acoplamiento, y se convierte en V_{div} gracias a la función $f(V_{bat})$. Dentro del microcontrolador esta señal es afectada por la calibración de las referencias positivas y negativas del conversor A/D que fijan los voltajes umbral máximo mínimo del canal de conversión.

Figura 4.14 Etapas que atraviesa la señal de carga de batería en el sistema



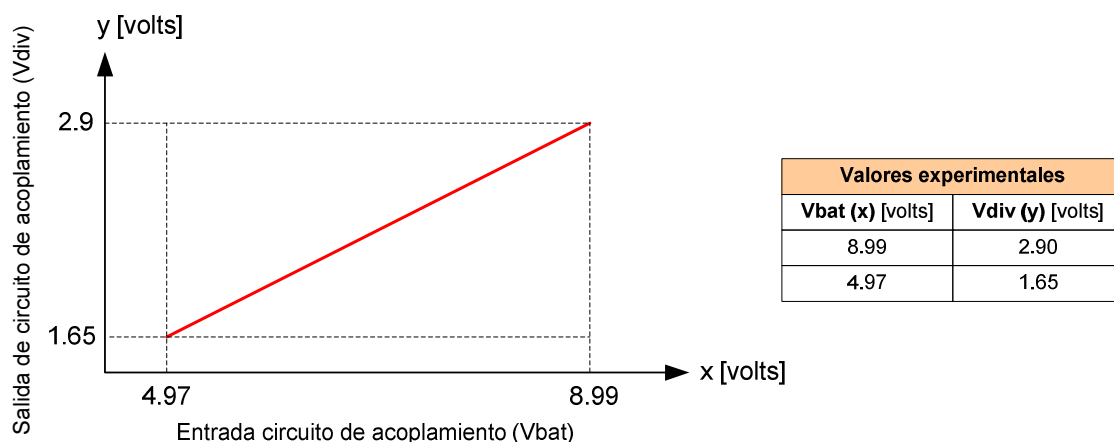
Es entonces que a través de la función $g(Vdiv)$, la variable $Vdiv$ pasa a ser N como variable discreta con una densidad de 10bits, que en dominio de software corresponde a un número entero que toma valores de 0 a 1023. Posteriormente la función $h(N)$ convierte estos números enteros en un número flotante V que contiene el voltaje de la batería similar ó igual a $Vbat$. Finalmente la función $i(V)$ convierte el valor V a valores puntuales de carga de batería, que en pantalla se presentarán como un dígito de 0 al 9, correspondiente a una carga del 0 al 100% respectivamente. Cuando la carga de la batería supera los 9V, el software presenta el carácter 'F' de *full* ó lleno en lugar del dígito 0 a 9.

A continuación se enumeran las funciones necesarias, antes mencionadas para obtener el nivel de carga de batería en pantalla, desde su conversión:

- Circuito de acoplamiento $f(Vbat)$
- Conversor A/D $g(Vdiv)$
- Relación voltaje $h(N)$
- Relación nivel $i(V)$

Circuito de acoplamiento $f(Vbat)$

Figura 4.15 Relación entre las señales de entrada y salida del circuito de acoplamiento



La función $f(V_{bat})$ representa la relación que existe entre el voltaje de la batería que se encuentra a la entrada del circuito de acoplamiento. Los valores puntuales para el cálculo de la función, fueron obtenidos experimentalmente mediante la inyección de voltaje en el circuito. El diseño circuito de acoplamiento se describe en el capítulo 5.

Según la gráfica 4.14, tenemos: La relación $V_{div} = f(V_{bat})$ es una relación lineal tipo $y = mx + b$, y se calcula como sigue:

$$m = \frac{2.9 - 1.65}{8.99 - 4.97} = 0.3109$$

$$b = y - mx = -(0.3109 \cdot 4.97) + 1.65 = 0.1046$$

$$y = 0.3109x + 0.1046$$

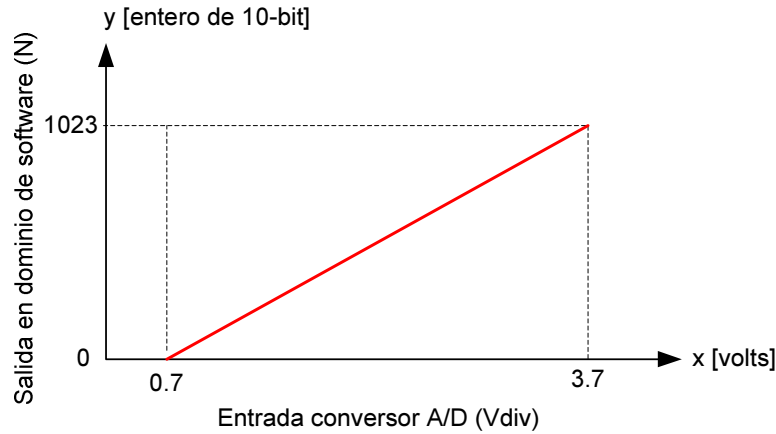
$$\rightarrow V_{div} = f(V_{bat}) = 0.3109 \cdot V_{bat} + 0.1046 \quad [Ec. 4.1]$$

Conversor A/D $g(V_{div})$

La función $g(V_{div})$ representa la relación que existe entre el voltaje de la salida del circuito de acoplamiento de la batería, y los valores numéricos de la salida del conversor A/D de 10 bits del microcontrolador.

Los umbrales de voltaje del conversor A/D, se ven afectados por la calibración del hardware de la tarjeta del prototipo. Dicha calibración obedece al diseño del circuito de entrada del canal de conversión externa que incluye un amplificador operacional. Este circuito de entrada fue diseñado e implementado en la tarjeta del prototipo GNOM-4550 para aplicaciones que requieran capturar una señal continua de voltaje, por ejemplo un osciloscopio ó analizador de espectro. El circuito de canal de conversión externa no está en el alcance de este proyecto por esta razón no se incluye su diseño en este documento; pero se definen los valores de umbral utilizados, y son:

- Umbral de alto nivel -> 3.7 voltios
 - Umbral de bajo nivel -> 0.7 voltios
-

Figura 4.16 Relación entre las señales de entrada y salida del conversor A/D

Donde: la relación $N = g(Vdiv)$ es una relación lineal tipo $y = mx + b$, y se calcula como sigue:

$$m = \frac{1023 - 0}{3.7 - 0.7} = 341$$

$$b = y - mx = 1023 - (341 \cdot 3.7) = -238.7$$

$$y = 341 \cdot x - 238.7$$

$$\rightarrow N = g(Vdiv) = 341 \cdot Vdiv - 238.7 \quad [Ec. 4.2]$$

Relación voltaje $h(N)$

La función $h(N)$ representa la relación que existe entre los valores numéricos de la salida del conversor A/D y los valores de voltaje que corresponden a la medida de la carga de la batería. Para encontrar el voltaje de la batería en términos de una variable dentro del programa, se reemplaza el valor de $Vdiv$ de la ecuación 4.1 en la ecuación 4.2, y se despeja $Vbat$ de la siguiente manera:

$$N = 341 \cdot (0.3109 \cdot Vbat + 0.1046) - 238.7$$

$$N = 106.0169 \cdot Vbat - 203.0314$$

$$Vbat = \frac{N + 203.0314}{106.0169}$$

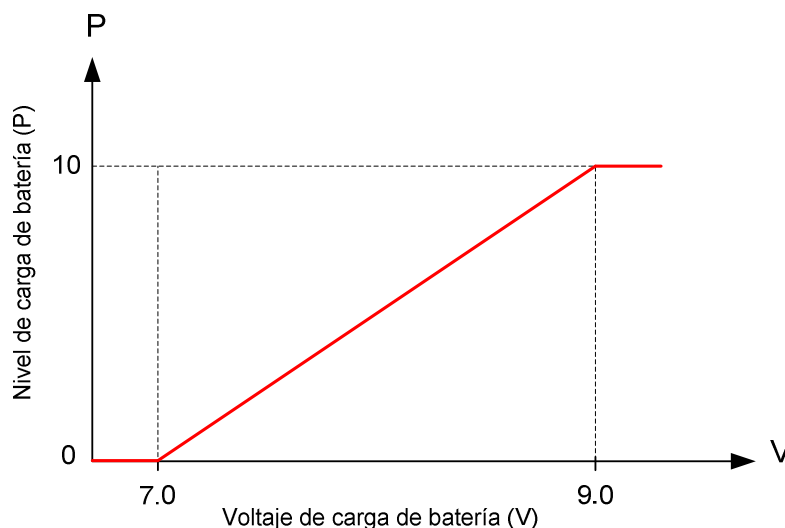
$$\rightarrow V = h(N) = \frac{N + 203.0314}{106.0169} \quad [Ec. 4.3]$$

Se puede observar que la variable V es igual a V_{bat} , es decir V en el dominio de software contiene el valor de voltaje de carga de batería V_{bat} como número flotante.

Relación nivel $i(V)$

La relación nivel es una función que relaciona un valor numérico en una escala de 0 al 10 que se observará en la pantalla GLCD, versus los valores de salida de la función $h(N)$. Cuando el voltaje de carga de la batería descienda de 7 voltios, se dirá que el nivel de carga es 0, mientras que si el valor de carga llega a los 9 voltios el nivel de carga será 10. Para el comportamiento descrito se observa la siguiente relación:

Figura 4.17 Relación entre el voltaje de batería y el nivel de carga



Donde: la relación $P = i(V)$ es una relación lineal tipo $y = mx + b$, y se calcula como sigue:

$$m = \frac{10 - 0}{9 - 7} = 5$$

$$b = y - mx = 10 - (5 * 9) = -35$$

$$y = 5x - 35$$
$$\rightarrow P = i(V) = 5 \cdot V - 35 \quad [Ec. 4.4]$$

Cabe señalar que en los intervalos de borde, la función es una línea continua. Entonces la relación completa quedara de la siguiente manera:

$$\text{Cuando } V < 7 \rightarrow P = 0$$

$$\text{Cuando } 7 \leq V \leq 9 \rightarrow P = 5 \cdot V - 35$$

$$\text{Cuando } V > 9 \rightarrow P = 10$$

La función $i(V)$ es continúa sin embargo en la programación del microcontrolador se muestra solamente la escala del 1 al 10 truncando la parte entera del resultado de la función. Cuando la escala llega al valor de 10, este valor es reemplazado con el caracter “F” de full o lleno, indicando que la carga de la batería está en su máximo nivel.

4.6 CONTROL DE TRANSFERENCIA DE ARCHIVOS

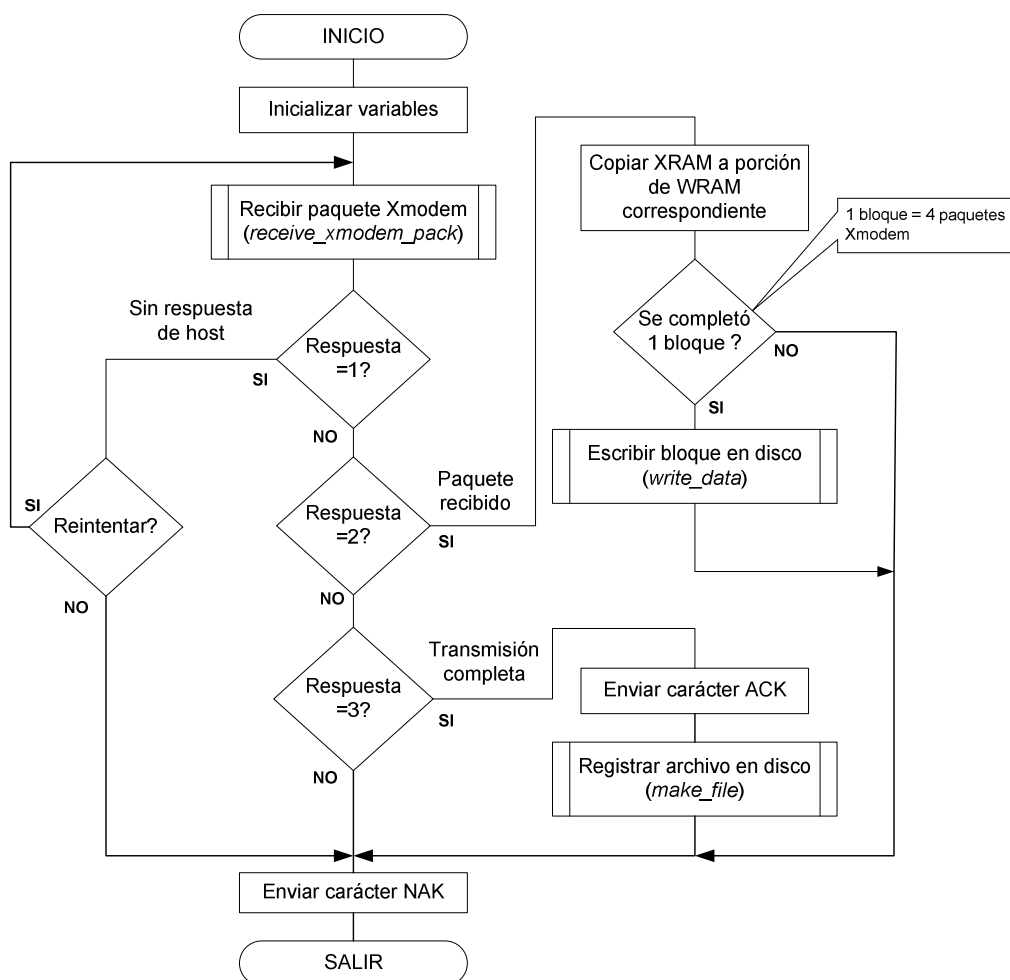
El grupo funcional “control de transferencia de archivos” es encargado de controlar la comunicación con la PC por medio del puerto serial usando el protocolo Xmodem, además de permitir la transferencia de archivos desde la PC hacia el prototipo GNOM-4550, almacenando y registrando los archivos en el disco SD. Este grupo funcional se encuentra implementado en las librerías de programa XMODEM.C y SDFDZ16.C Las funciones principales que realiza este grupo funcional son:

- Control de protocolo Xmodem
 - Transferencia y almacenamiento de información
 - Registro de archivo en disco
-

4.6.1 Control de protocolo Xmodem

El control de protocolo Xmodem se realiza principalmente a través de una función llamada: “*receive_external_file*”, la cual controla e interpreta el protocolo Xmodem utilizando el puerto serial RS-232. EL diagrama de bloques se muestra en la figura anterior.

Figura 4.18 Diagrama de flujo simplificado de la función “*receive_external_file*”



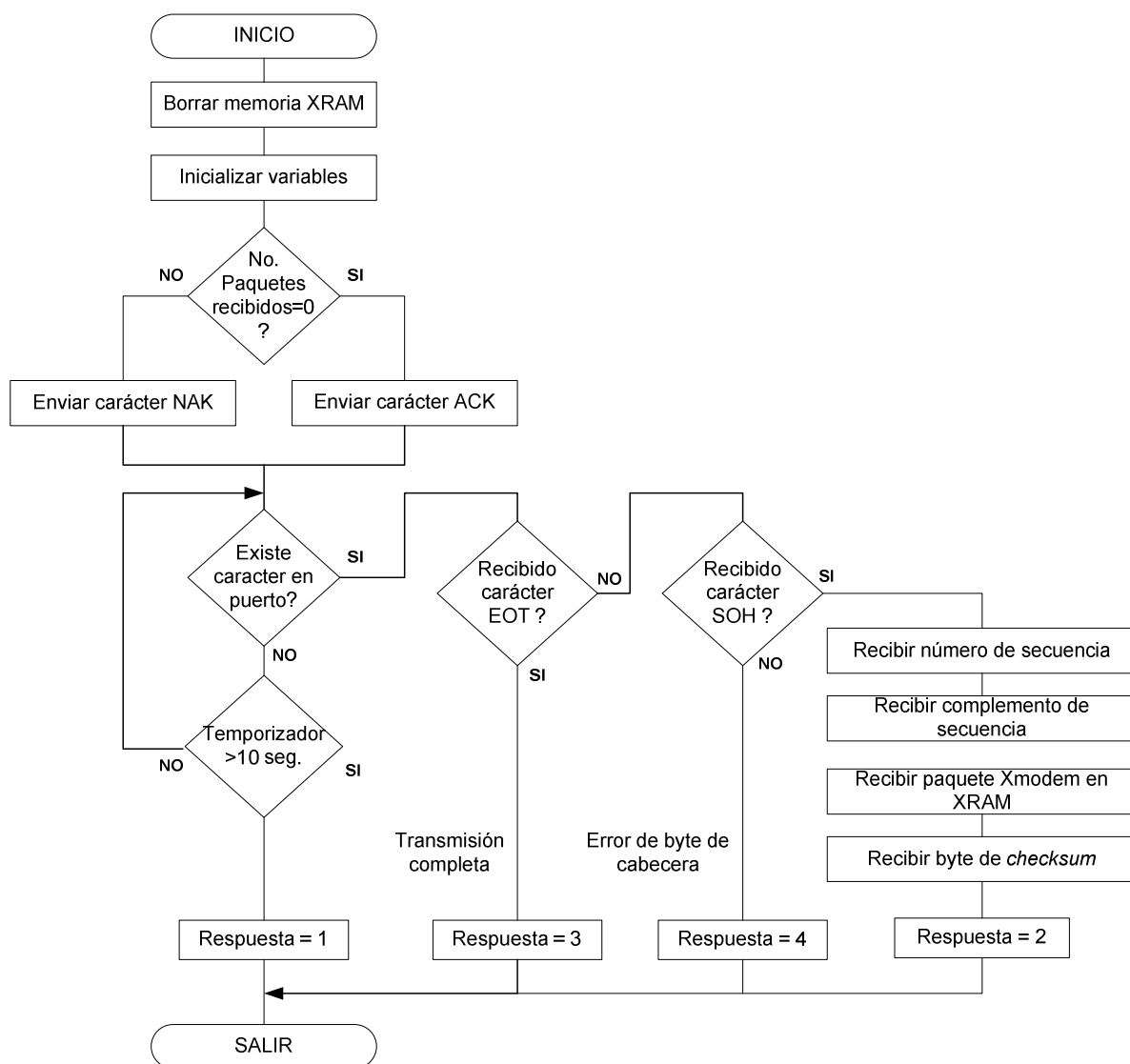
4.6.2 Transferencia y almacenamiento de información

La transferencia y el almacenamiento de la información en el medio físico, se realiza a través de dos funciones principales: *receive_xmodem_pack* y *write_data*,

- ***receive_xmodem_pack***

La función de recepción de paquetes Xmodem “*receive_xmodem_pack*” es la responsable de capturar el paquete de datos Xmodem y mantener el control de los mensajes del protocolo. La función se sirve de un contador de paquetes para el control interno. Los caracteres especiales del protocolo Xmodem como: EOT, SOH, ACK, NAK se explica en el capítulo 2.

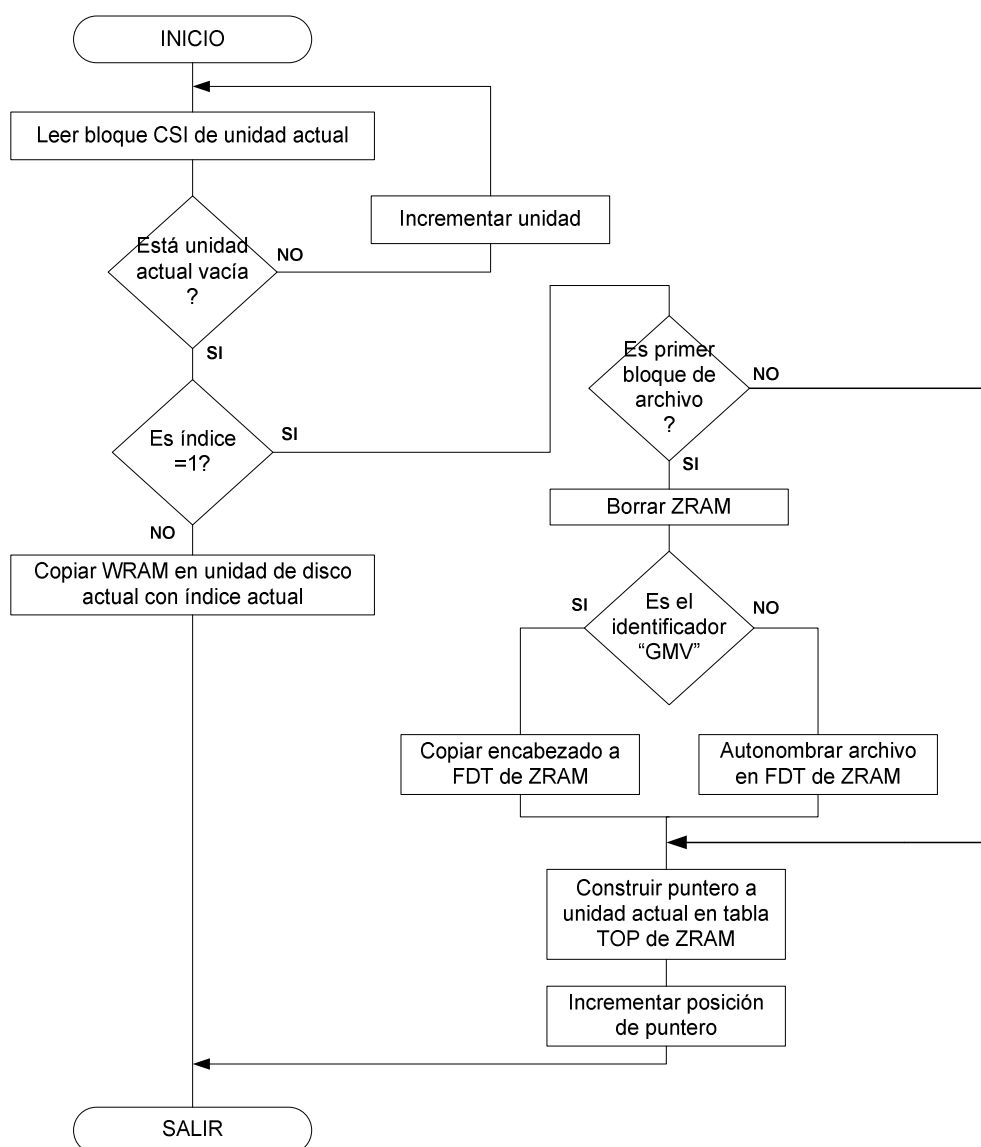
Figura 4.19 Diagrama de flujo simplificado de la función “*receive_xmodem_pack*”



- ***write_data***

La función de escritura de datos en el disco SD “*write_data*” es la responsable de buscar unidades vacías en el disco SD, copiar el bloque de datos desde WRAM hacia cada uno de los elementos de cada unidad, utilizando tantas como sean necesarias dependiendo de la longitud del archivo recibido. Esta función también es la responsable de construir gran parte del bloque FDB que corresponderá al archivo en proceso de construcción.

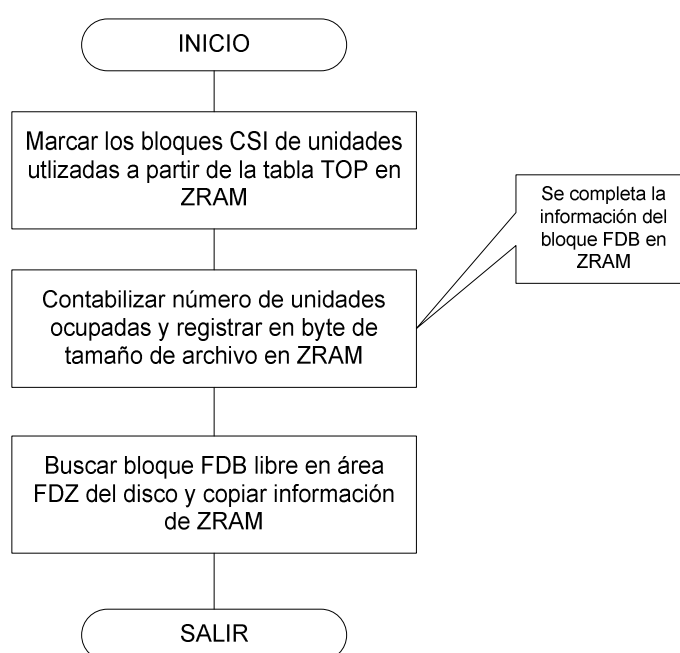
Figura 4.20 Diagrama de flujo simplificado de la función “*write_data*”



4.6.3 Registro de archivo en disco

El registro del archivo en el disco ó memoria SD se realiza por medio de la función “*make_file*” que se encarga de completar y escribir en el disco el bloque FDB que servirá para registrar en la memoria, la existencia del archivo transferido; a partir de la información que reside en la memoria ZRAM (ver capítulo 3). La estructura de la función “*make_file*” se observa en la siguiente figura.

Figura 4.21 Diagrama de flujo simplificado de la función “*make_file*”



4.7 CONTROL DE NAVEGACIÓN

El grupo funcional “control de navegación” es el corazón del prototipo, pues constituye el software que se encarga de la tarea de navegación GPS propiamente dicha, y se encuentra implementado mayormente en la librería NAVCTRL.C; cuyas funciones principales son las siguientes:

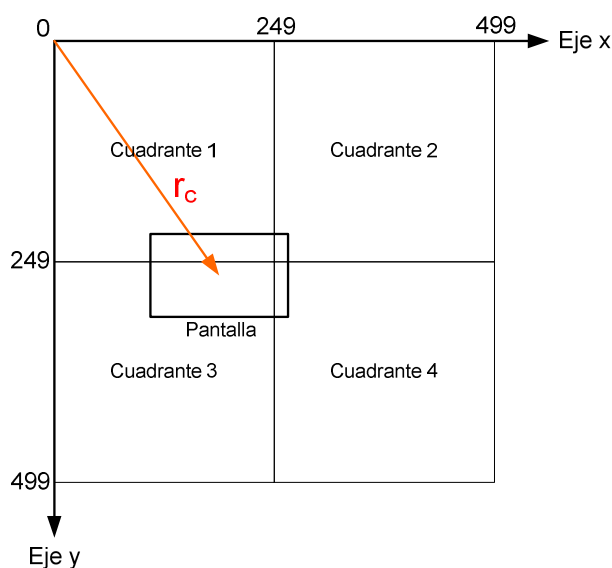
- Funcionamiento del núcleo de navegación
- Acceso a archivos (pre-lectura)

- Procesamiento de archivos (lectura)
- Tratamiento de datos GPS
- Control de cursor

4.7.1 Núcleo de navegación

El núcleo de navegación está formado por un sistema de cuatro cuadrantes ó unidades virtuales de operación gráfica de 250 x 250 píxeles cada una, donde se dibujarán cuatro zonas de un mapa digital correspondientes a una área geográfica determinada. Sobre estos cuatro cuadrantes se desplaza una ventana ó pantalla que corresponde al área visual del LCD gráfico de 128 x 64 píxeles, cuya posición es controlada mediante el vector bidimensional r_c .

Figura 4.22 Cuadrantes del control de navegación y área de pantalla LCD



A cada cuadrante se le asigna un grupo de datos que corresponden a una zona de un archivo de mapa digital. Este grupo de datos forma parte del registro de navegación NAVREG, y son los siguientes:

- 1) Número de zona
- 2) Datos de longitud: grados, minutos, dirección (W ó E)

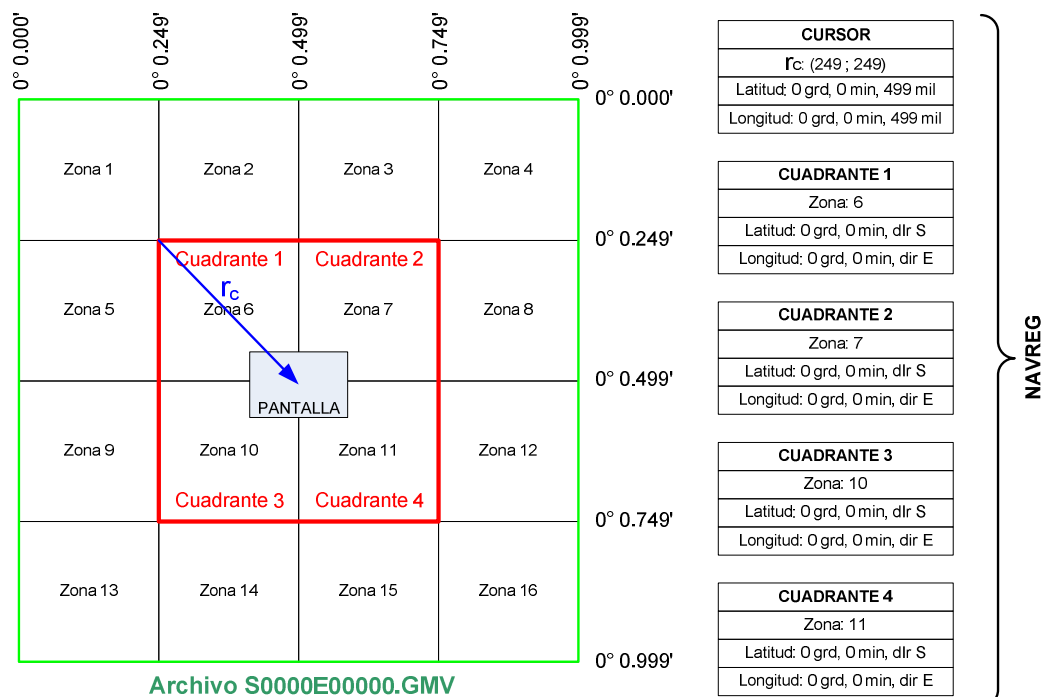
3) Datos de latitud: grados, minutos, dirección (N ó S)

El cursor se encuentra en la mitad de la pantalla LCD (64 x 32 pixel), cuya posición dentro del sistema de cuadrantes es determinada por medio del vector bidimensional r_c . Al este cursor están relacionados otro conjunto de datos que también forman parte del registro de navegación (NAVREG), que son:

- 1) Posición del cursor (vector r_c): r_{cx} , r_{cy}
- 2) Longitud del cursor: grados, minutos, milésimas
- 3) Latitud del cursor: grados, minutos, milésimas

Para identificar la dirección (hemisferios), las coordenadas del cursor son variables con signo; es decir, cuando el cursor se encuentre en los hemisferios norte (N) y este (E), las variables son números positivos; mientras que los hemisferios sur (S) y oeste (W), son representados por variables con números negativos.

Figura 4.23 Inicialización del núcleo de navegación y el registro NAVREG



La inicialización del núcleo de navegación se realiza al ingresar al navegador por primera vez, especificando las coordenadas de un área geográfica que cubre un archivo GMV; es decir es necesario indicar los grados y minutos de latitud y longitud; además de la dirección de dichas coordenadas. La función de inicialización posicionará automáticamente la pantalla por medio del vector r_c en la mitad del sistema de cuadrantes ($r_{cx}=249$, $r_{cy}=249$), y el sistema de cuadrantes en las zonas correspondientes a la mitad del área geográfica que cubre el archivo (ver figura 4.22).

Por ejemplo, la unidad está programada para inicializar en: latitud: 0 grados, 0 minutos, S (hemisferio sur), y longitud: 0 grados, 0 minutos, E (hemisferio este), donde el controlador calculará los datos del registro de navegación correspondiente a cada cuadrante y seguidamente también calcula los datos del cursor.

Cuando la posición de la ventana excede los límites del cuadrante se produce un intercambio de zona, es decir el controlador procederá a determinar la nueva zona que le corresponde a cada cuadrante, además de re posicionar la ventana dentro del sistema de cuadrantes. Los cambios de zona se pueden dar entre un mismo archivo ó entre varios archivos, dependiendo del área geográfica donde se encuentre el cursor.

4.7.2 Acceso a archivos (pre-lectura)

La pre-lectura de archivos de navegación es el proceso por el cual un archivo GMV es solicitado, localizado dentro del disco SD, y preparado para el siguiente proceso que es la lectura de archivo. Las funciones que se realizan en la pre-lectura de archivos GMV son:

- 1) Petición de archivo
 - 2) Interpretación de nombre de archivo
 - 3) Búsqueda de archivo en disco
-

4) Lectura de encabezado y construcción de tabla NFDI

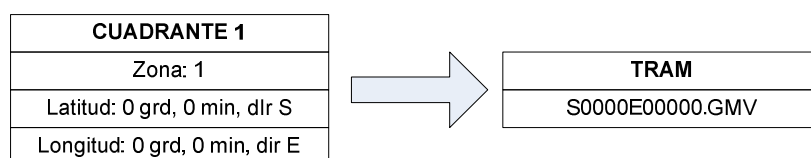
Petición de archivo

La petición de archivo se realiza con cada actualización de datos del registro de navegación (NAVREG); es decir con cada cambio de zona del núcleo de navegación.

Interpretación de nombre de archivo

El proceso de interpretación de nombre de archivo, toma los datos del registro de navegación y los convierte a nombre de archivo en forma de caracteres ASCII, y lo deposita en la memoria TRAM. Cada cuadrante del núcleo de navegación utiliza esta función para buscar el archivo que le corresponde (ver siguiente figura).

Figura 4.24 Interpretación de nombres de archivo



Búsqueda de archivo en disco

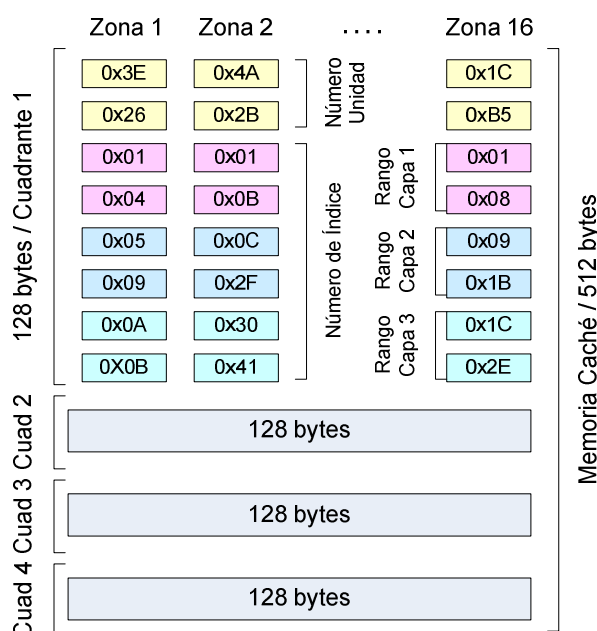
Este proceso utiliza la información de TRAM y realiza una búsqueda del archivo comparando la cadena de caracteres con el nombre de archivo de cada bloque FDB ocupado en el disco. Si la información es coincidente, se procede con el siguiente paso, caso contrario el sistema despliega un mensaje indicando el archivo faltante.

Lectura de encabezado y construcción de tabla NFDI

Una vez ubicado el archivo GMV, el CPU se encarga de leer el encabezado del archivo y procesarlo de tal manera que genere una tabla NFDI en la memoria ZRAM. Una tabla NFDI (*Navigation File Description Indicators*) ó tabla de indicadores de descripción de archivo de navegación, es una tabla de punteros de memoria que ayudan al controlador a saber el rango de direcciones lógicas donde se encuentra cierto archivo en el disco SD, dividido por zona y capas.

La tabla NFDI es construida a partir del procesamiento de la información del encabezado del archivo. Los punteros de memoria de la tabla NFDI son diferentes de los del encabezado porque están basados en el arreglo lógico del sistema de archivos, es decir dependen de cómo y donde se hayan alojado estos archivos dentro del disco (memoria SD).

Figura 4.25 Estructura de la memoria caché formada por cuatro tablas NFDI



Debido a las limitaciones inherentes del microcontrolador, no todas las configuraciones de archivo GMV son susceptibles de lectura en la unidad

GNOM-4550. Por esta razón la tabla NFDI es construida solamente para archivos con 3 capas y un tamaño de 1 unidad (30KB) por zona, más un tamaño de encabezado de 1 bloque (512 bytes); es decir, el tamaño máximo de archivo se calculará como: $(30\text{KB} \times 16 \text{ zonas}) + 0.5\text{KB} = 480.500 \text{ bytes}$ ó 481KB.

En base a esta configuración de archivo, el controlador construirá 4 tablas NFDI para los cuatro cuadrantes del sistema de navegación, con 128 bytes cada una; donde la memoria ZRAM total es de $128 \times 4 = 512 \text{ bytes}$. El contenido de la tabla se forma de un arreglo de 16 columnas que representan las 16 zonas del mapa, con 8 bytes cada una. En los dos primeros bytes de la columna se encuentra la dirección lógica del disco ó número de unidad que contiene los datos de una zona. En el resto de elementos de la columna, se alojan los números de índice que separan la información de zona en capas (ver figura 4.25).

En el ejemplo de la figura 4.25, se observa que en la primera columna que corresponde a los datos de la zona 1, los dos primeros bytes corresponden a la dirección lógica de la unidad donde se encuentran los datos de dicha zona, siendo MSB el primer byte y LSB el segundo byte, que forman un puntero de 16 bits.

$$0x3E \text{ recorrido 8 bits a la izquierda} = 0x3E00 + 0x26 = 0x3E26$$

$$\rightarrow \text{puntero a unidad 15910}$$

Del tercer byte al octavo de cada columna se encuentran los rangos de índices de los datos de cada capa de una determinada zona. Es decir en el ejemplo de la primera columna (Zona 1), la capa 1 estará en los índices 1 al 4, la capa 2 estará en los índices 5 al 9, y la capa 3 del 10 al 11.

Con las tablas NFDI, el controlador agiliza el proceso de lectura de datos de los archivos del mapa digital; velocidad que resulta necesaria para obtener un buen rendimiento de trabajo del navegador GPS.

4.7.3 Procesamiento de archivos (lectura)

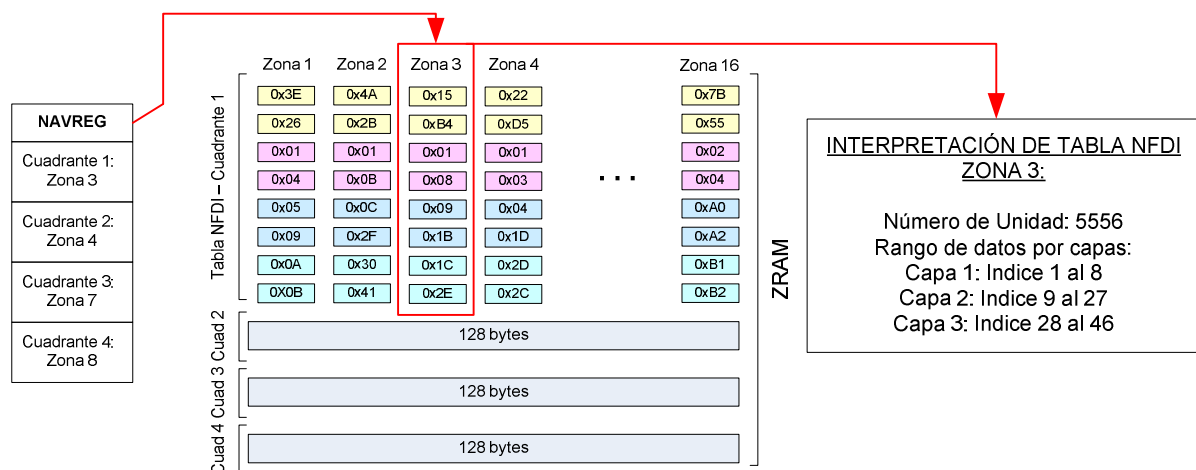
En este proceso el controlador lee e interpreta la información de las tablas NFDI contenidas en ZRAM, que utiliza para extraer del disco SD, la información del cuerpo del archivo (ó archivos), y permitir que el graficador (ó software de graficación) transforme dicha información en un mapa digital en pantalla.

Las funciones principales que desempeña la lectura de un archivo son:

- 1) Interpretación de tablas NFDI
- 2) Lectura de cuerpo de archivo
- 3) Graficación

Interpretación de tablas NFDI

Figura 4.26 Ejemplo de interpretación de tabla NFDI



El controlador lee la memoria ZRAM con cada actualización de zona en el núcleo de navegación. Esta función, procesa las tablas 4 tablas NFDI, que corresponden a cada cuadrante, y ubica mediante los punteros de memoria, la posición de los bloques de memoria dentro del disco donde se encuentra la información del cuerpo del archivo. La función de interpretación de tablas NFDI

utiliza también la información del registro NAVREG para ubicar la zona que le corresponde a cada cuadrante del núcleo de navegación (ver figura 4.25).

Lectura de cuerpo de archivo

Una vez que el controlador ha interpretado una tabla NFDI, procede a leer del disco los datos correspondientes al cada cuadrante con su respectiva zona. La lectura utiliza las funciones del controlador de disco SD, cuya información se deposita por bloques en la memoria RRAM.

La lectura de cada capa depende de la habilitación de los *flags* correspondientes a la configuración “*Layer Enable*” del submenú de navegación. Dependiendo de esta configuración es posible que el usuario observe el mapa digital en pantalla omitiendo o mostrando ciertas capas.

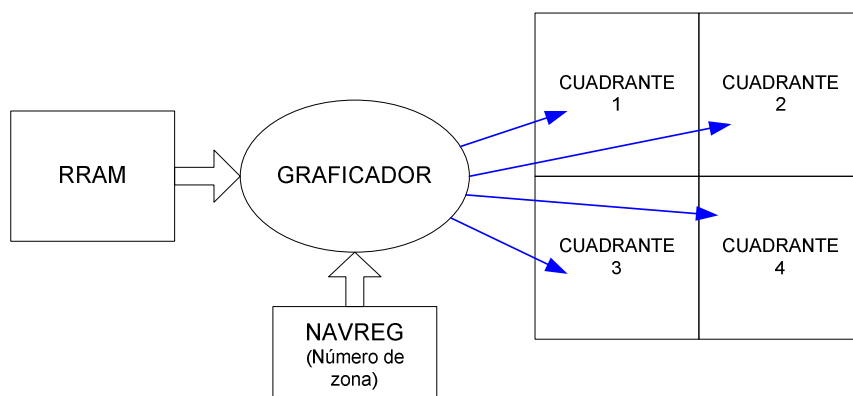
Figura 4.27 Pantalla de configuración de flags en la unidad GNOM-4550



Graficación

El graficador (función de graficación), toma un bloque de datos cada vez desde RRAM, identifica las sentencias compuestas de comandos y argumentos, y los grafica según el caso, en un cuadrante del núcleo de navegación.

La función de graficación utiliza los datos del registro NAVREG para ubicar la zona a la que corresponde a un cuadrante determinado.

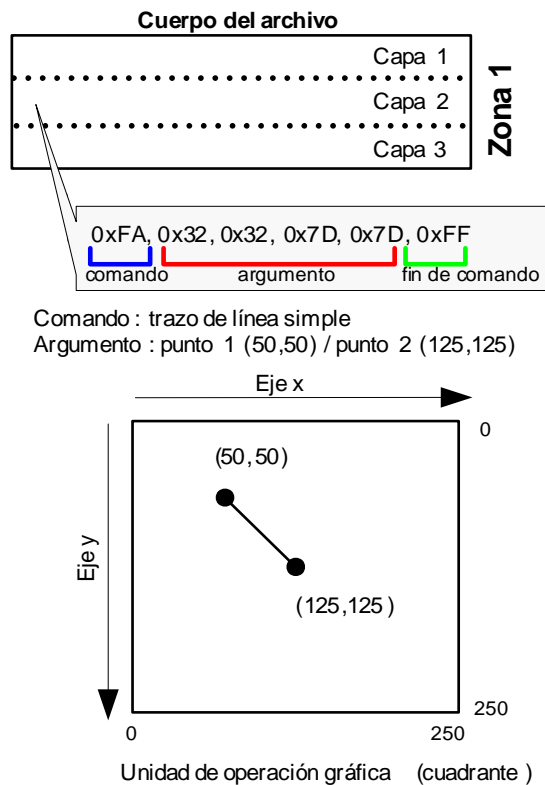
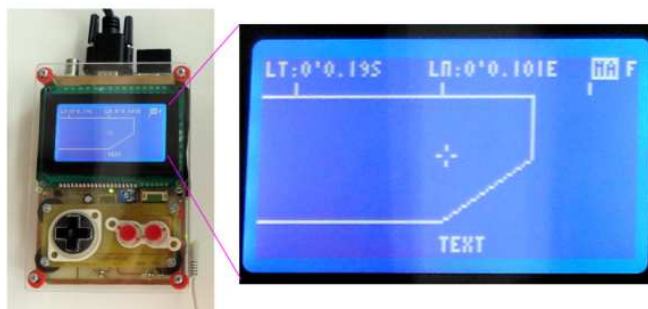
Figura 4.28 Estructura de la función de graficación

La operación de graficación se realiza cada vez que una actualización de zona es requerida ó iniciada por algún componente de programa. Esta actualización se realiza cada vez que la ventana rebaza los límites de los cuadrantes del núcleo de navegación.

La interpretación del contenido del cuerpo del archivo se realiza en base a la identificación de los datos entre comandos y coordenadas. En la figura 4.29 se ilustra un ejemplo de interpretación de datos del cuerpo del archivo GMV para formar una línea.

En la modalidad GPS, el controlador del sistema utiliza una función de captura de trama GPS y conversión de datos hacia un registro interno que guarda la información de la posición de usuario en latitud y longitud, velocidad, hora, y fecha.

Para ubicar la posición de usuario en pantalla y relacionarla con el mapa digital; la unidad GNOM-4550 utiliza un proceso de igualación de datos y corrimiento de cursor. Con este proceso los registros de navegación contienen ahora la información relacionada a la posición de usuario; de esta manera se pueden realizar las funciones de pre-lectura y lectura de mapa, del área geográfica donde se encuentra el receptor GPS.

Figura 4.29 Ejemplo de interpretación de datos del cuerpo de archivo**Figura 4.30 Ejemplo de trazo de líneas y texto en la unidad GNOM-4550**

4.7.4 Tratamiento de datos GPS

El receptor GPS utilizado (*GlobalSat BR-355*), es un receptor con interfaz serial asincrónica, que envía datos por medio del protocolo NMEA0183; que son capturados por medio del puerto serial del prototipo GNOM-4550.

El mensaje GPS utilizado es la trama GPRMC ó Información Mínima de Navegación (ver capítulo 2), donde se encuentran todos los datos necesarios para lograr los objetivos planteados.

Captura de trama GPS

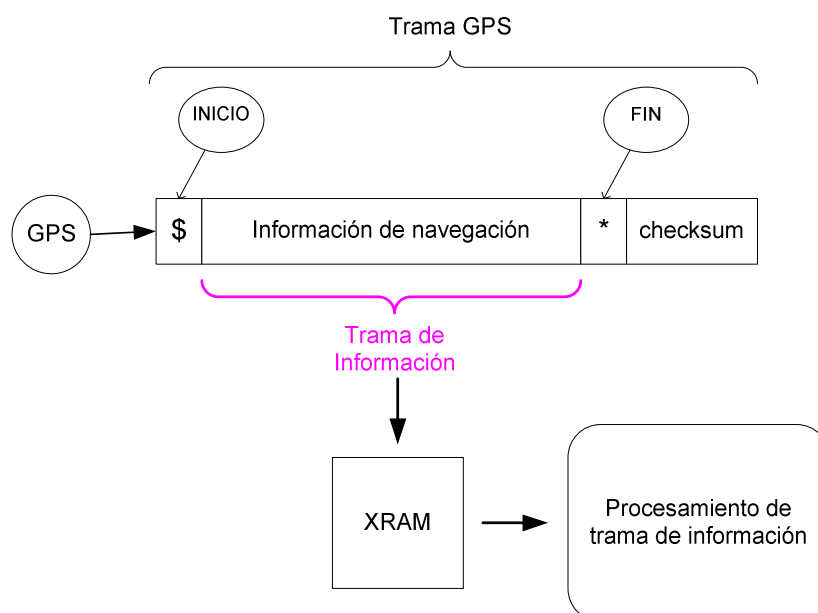
Analizando el tiempo de bit para el protocolo NMEA0183, tenemos que:

$$\begin{aligned}Tasa\ de\ bits &= 4800bps \\Tiempo\ de\ bit &= \frac{1}{Tasa} = \frac{1}{4800bps} = 208.3\mu S\end{aligned}$$

Para capturar las tramas GPS, es necesaria una captura carácter por carácter en el puerto serial utilizando una interrupción de temporización menor al tiempo de bits calculado. El periodo de interrupción debe ser lo suficientemente pequeño como para capturar sin pérdida cada carácter ASCII, pero sin que llegue a interponerse con la normal ejecución del programa principal. De esta manera, mediante experimentación se obtuvo un periodo de interrupción óptimo de 42.6 us.

Con el fin de procesar una trama completa NMEA0183, es necesario identificar tanto el inicio como el fin de trama. Un denominador común al inicio de cada trama es el carácter "\$", que señalará el inicio de cualquier tipo de trama. Debido a que existen distintos tipos de trama con diferente información, la longitud es variable; y un conteo de caracteres sería impreciso para la identificación del final de trama. En su lugar se optó por localizar el carácter "*" que corresponde al identificador de *checksum* al final de todas las tramas.

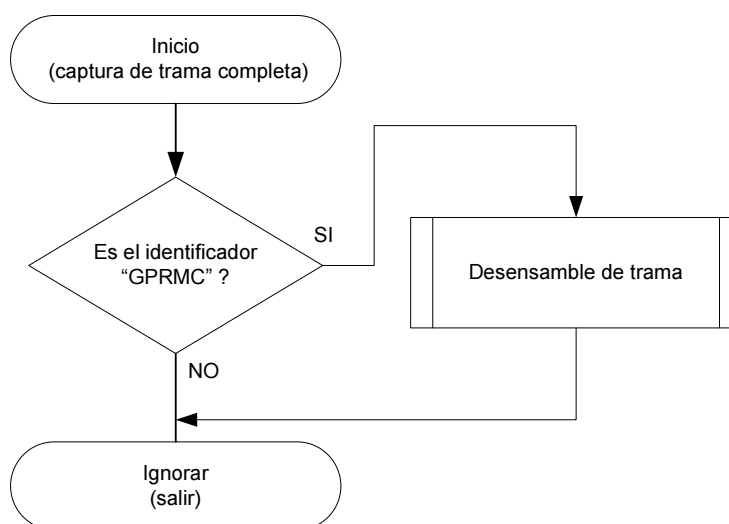
Al identificar cada inicio y fin de trama, esta se la puede almacenar en memoria para que un proceso de detección posterior pueda detonarse entre dichas tramas (ver siguiente figura).

Figura 4.31 Diagrama de identificación de trama GPS

La sentencia que contiene los datos GPS entre los caracteres “\$” y “*” se denomina trama de información. Esta trama de información es alojada en la memoria XRAM para su procesamiento. La longitud máxima de una trama según el protocolo NMEA0183 es de aproximadamente 80 caracteres. El tiempo que existe entre cada fin de trama y comienzo de la siguiente, el microcontrolador realiza el procesamiento de trama de información.\

Procesamiento de trama de información

La trama de información que se utiliza es la trama GPRMC ó denominada también trama de información básica de navegación. El procesamiento de trama toma los datos de la memoria XRAM, y procede a identificar primeramente el tipo de trama, comparando si los primeros caracteres son coincidentes con “GPRMC”. Si encuentra similitud procede al desensamble de la trama; caso contrario la ignora.

Figura 4.32 Diagrama de procesamiento de trama de información

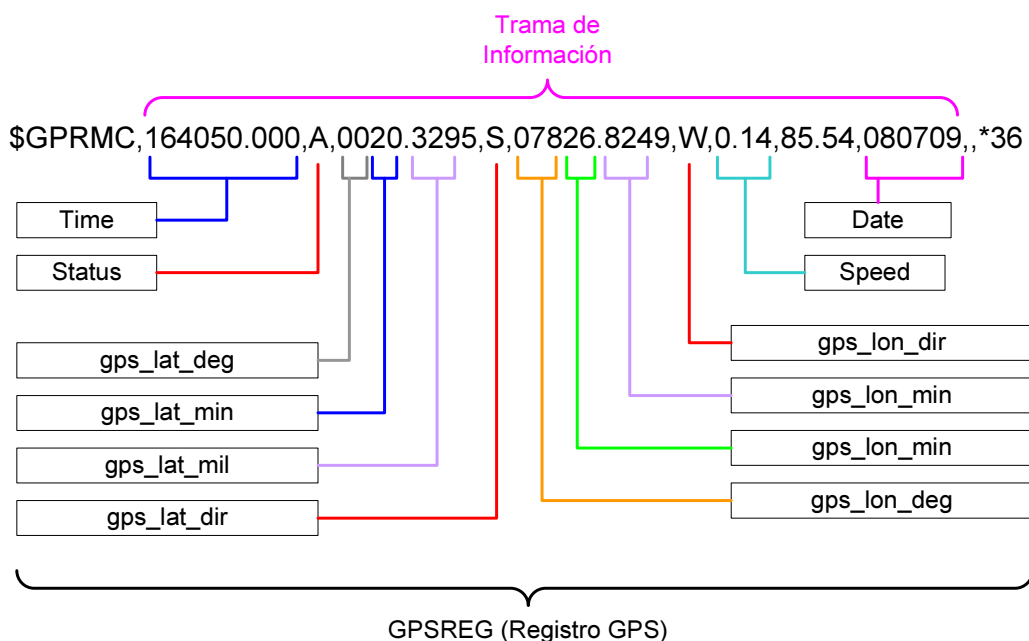
El desarmado o desensamble de la trama de información GPRMC, contempla la separación de cada elemento y la conversión de cadenas de caracteres a valores numéricos, que se almacenan en el registro GPS (GPSREG).

El registro GPSREG contiene 12 variables donde se almacena la información de posicionamiento de usuario proveniente de la trama GPRMC. La explicación detallada de cada instancia de la trama se encuentra en el capítulo 2. A continuación se explican las 12 variables del registro GPSREG.

- Status: caracter de estado del receptor.
 - Time: grupo de 4 caracteres donde se almacena la hora (hh:mm)
 - Date: grupo de 6 caracteres donde se almacena la fecha (ddmmaa)
 - Speed: entero donde se almacena la velocidad en Km/h
 - Gps_lon_deg: entero donde se almacenan los grados longitudinales
 - Gps_lon_min: entero donde se almacenan los minutos longitudinales
 - Gps_lon_mil: entero donde se almacenan las milésimas de minuto longitudinal
 - Gps_lon_dir: caracter donde se almacena el hemisferio longitudinal
-

- Gps_lat_deg: entero donde se almacenan los grados latitudinales
- Gps_lat_min: entero donde se almacenan los minutos latitudinales
- Gps_lat_mil: entero donde se almacenan las milésimas de minuto latitudinales
- Gps_lat_dir: caracter donde se almacena el hemisferio latitudinal

Figura 4.33 Desensamble de trama GPRMC en las variables de registro GPSREG



El controlador utiliza los datos del registro GPSREG dentro de su sistema para compararla con los datos del mapa digital y encontrar la posición de usuario. Debido a que en términos de hora y fecha no se realiza conversión de datos, la información proporcionada corresponde al tiempo universal coordinado ó UTC.

CAPÍTULO 5

DISEÑO Y CONFIGURACIÓN DE HARDWARE

-
- 5.1 Estructura general del sistema**
 - 5.2 Núcleo del sistema (microcontrolador PIC18F4550)**
 - 5.3 Periféricos de entrada / salida**
 - 5.4 Fuente de poder**
-

5.1 ESTRUCTURA GENERAL DEL SISTEMA

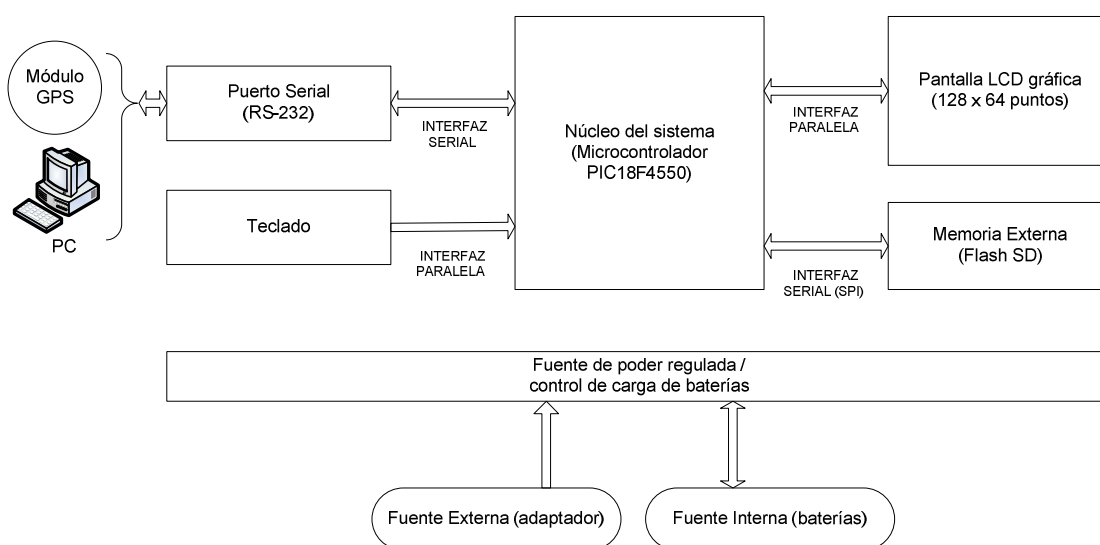
El sistema se implementa sobre una plataforma basada en el microcontrolador PIC18F4550 de Microchip®, que es el núcleo del dispositivo donde se encuentra el software de control y procesamiento.

Existen cuatro periféricos en el sistema además de una fuente de poder (ver figura 5.1). Cada uno de estos periféricos se comunica con la unidad central de proceso por medio de los módulos de comunicación disponibles en el microcontrolador.

Las interfaces paralelas hacen uso de las líneas directas de entrada / salida, mientras que las interfaces seriales utilizan el módulo MSSP (*master synchronous serial port*) en modalidad SPI para la comunicación con la tarjeta SD; además de un controlador de software para el puerto serial RS-232.

La fuente de poder utiliza un canal del conversor análogo / digital del microcontrolador, con el fin de medir el nivel de carga de la batería, además se dispone de una circuitería de recarga con regulador de corriente constante ajustable.

Figura 5.1 Estructura en bloques del sistema



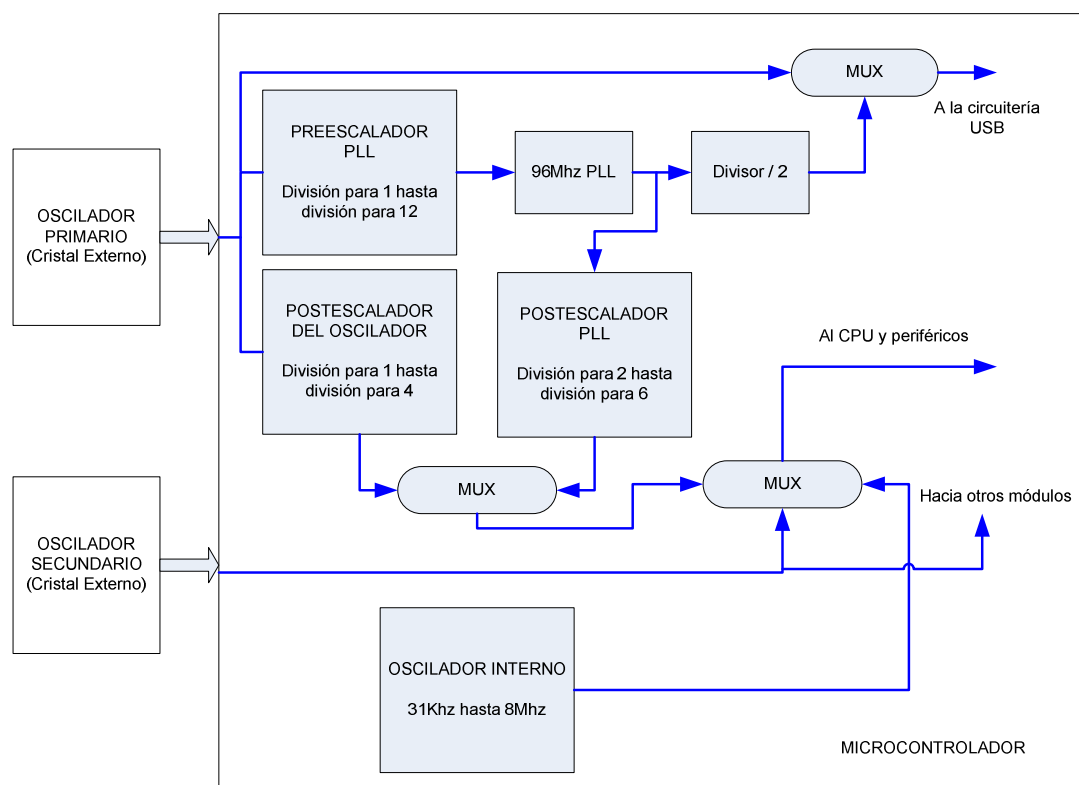
5.2 NÚCLEO DEL SISTEMA (MICROCONTROLADOR PIC18F4550)

El microcontrolador PIC18F4550 debe operar con la configuración de reloj que provea la frecuencia más alta posible dentro de los parámetros tolerables del resto de componentes del sistema, con el fin de lograr obtener una velocidad de procesamiento lo suficientemente rápida para la aplicación deseada.

5.2.1 Estructura y configuración del reloj

El PIC18F4550 posee una amplia gama de configuraciones de reloj para modificar la fuente de oscilación y la frecuencia de trabajo de cada módulo que compone el microcontrolador, ya sea con cristal resonador externo o generado internamente. A continuación se muestra el diagrama simplificado de la arquitectura interna de reloj.

Figura 5.2 Estructura del reloj interno del PIC18F4550

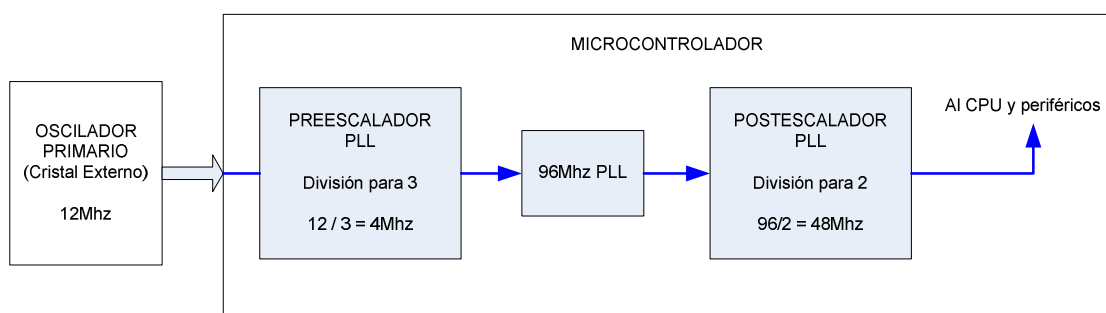


Como se observó en la figura 5.2, existen tres fuentes de oscilación para el sistema: oscilador primario, oscilador secundario y oscilador interno. El oscilador primario puede funcionar directamente o a través de un PLL (*phase locked loop*) ó anillo de enganche de fase, que genera una frecuencia de oscilación de 96Mhz y sirve de base de tiempo para generar varias frecuencias de reloj por medio de un post-escalador, siendo capaz el CPU de trabajar con una frecuencia máxima de 48Mhz.

El oscilador secundario trabaja directamente con el CPU y otros módulos internos del microcontrolador. En el modo de oscilador interno, se pueden obtener frecuencias de trabajo de hasta 8Mhz.

En el presente diseño se escogió un cristal resonador de 12Mhz que excitará el PLL interno para generar la frecuencia interna de 96Mhz, usando el puerto de oscilación primario. El pre-escalador debe dividir para 3 la frecuencia de oscilación natural del cristal resonador para obtener 4Mhz a la entrada del PLL, que es la especificada por el fabricante; y así poder obtener a la frecuencia interna de 96Mhz en el PLL.

Figura 5.3 Configuración del PLL interno del microcontrolador



Mediante pruebas realizadas se obtuvo que todos los componentes del sistema pueden trabajar sin problemas a la frecuencia máxima de reloj del microcontrolador de 48Mhz.

5.3 PERIFÉRICOS DE ENTRADA / SALIDA

La comunicación con los periféricos de entrada y salida del prototipo se la realiza por medio de comunicación serial y paralela desde y hacia el microcontrolador. A continuación se detalla cada uno de los componentes del sistema y su configuración de puertos.

5.3.1 Configuración del puerto de display GLCD

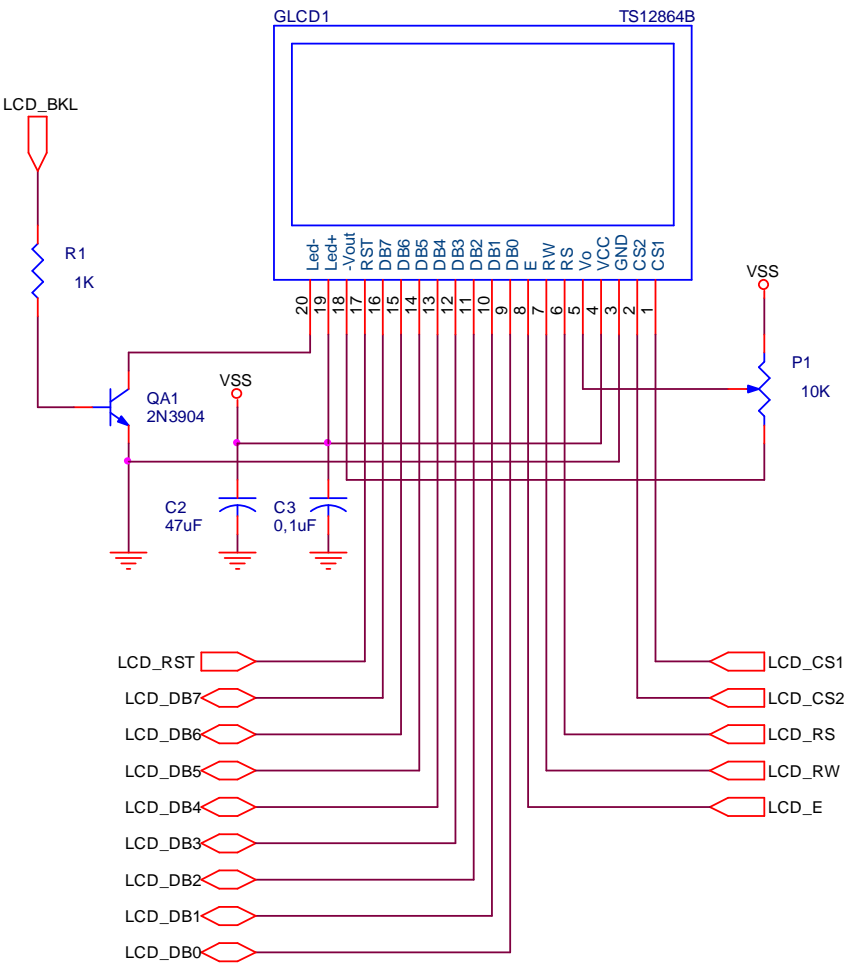
El display LCD de gráficos ó GLCD, se comunica desde y hacia el microcontrolador por medio de una interfaz paralela donde existen tres buses principales:

- Bus de datos
- Bus de control
- Bus de alimentación

El bus de datos tiene una densidad de 8 bits y es un bus de comunicación bidireccional con voltajes de trabajo TTL. Se compone de 8 líneas desde DB0 hasta DB7. El bus de control posee 5 señales que permiten el control del display, y por último el bus de alimentación que se compone de 6 líneas para la polarización y control de contraste del LCD.

El control de contraste del display se realiza variando el voltaje en el pin VO por medio de un potenciómetro entre +5V y -15V, que provienen de la fuente de alimentación general y el potencial generado por la circuitería interna del LCD respectivamente. El control de brillo de la pantalla se lo realiza por medio del control de la intensidad luminosa de los led del GLCD a través de una salida PWM del microcontrolador.

Figura 5.4 Diagrama de conexiones del display GLCD



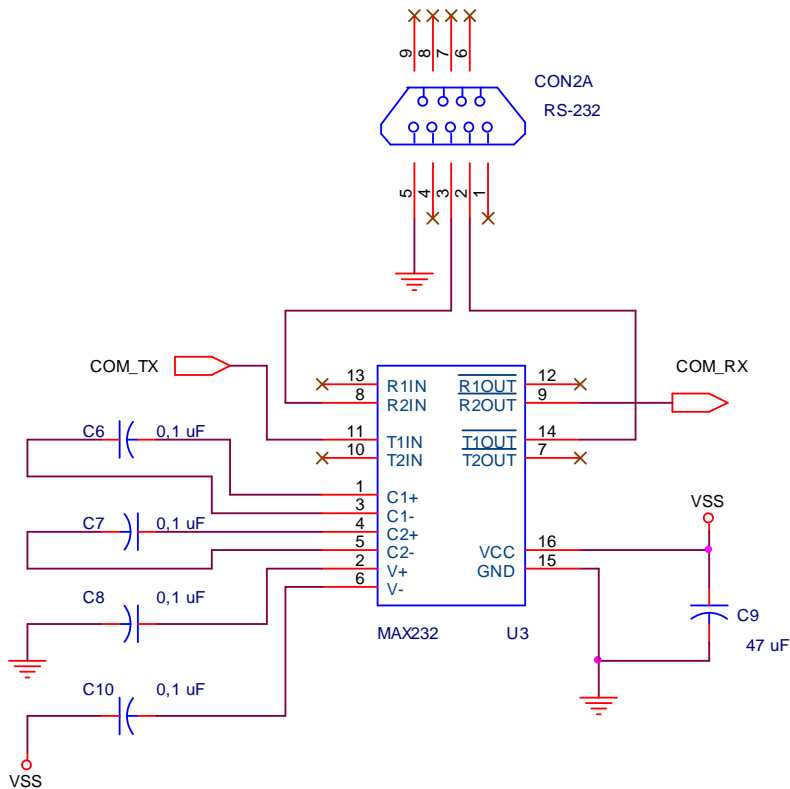
Líneas desde el display	#PIN	Nombre	Descripción	Líneas hacia el microcontrolador
Bus de Datos				
DB0	9	Data Bit 0	Bit menos significativo del bus de datos (LSB)	RD0
DB1	10	Data Bit 1	.	RD1
DB2	11	Data Bit 2	.	RD2
DB3	12	Data Bit 3	.	RD3
DB4	13	Data Bit 4	.	RD4
DB5	14	Data Bit 5	.	RD5
DB6	15	Data Bit 6	.	RD6
DB7	16	Data Bit 7	Bit más significativo del bus de datos (MSB)	RD7
Bus de Control				
LCD_CS1	1	Chip Select 1	Direcciona el primer controlador de la pantalla	RB6
LCD_CS2	2	Chip Select 2	Direcciona el segundo controlador de la pantalla	RB7
LCD_DI	6	Data / Instruction	Selecciona entre datos e instrucciones	RB2
LCD_RW	7	Read / Write	Selecciona entre lectura y escritura de datos	RB3
LCD_E	8	Enable	Habilitación del controlador común	RB4
LCD_RST	17	Reset	Reinicializa el display LCD	RB5

Bus de Alimentación				
GND		-	Tierra de señal y alimentación	-
VCC	4	+5V	Alimentación de 5V	-
VO	5	-	Control de contraste	-
-Vout	18	-	Salida de voltaje negativo para contraste	-
Led+	19	-	Polarización positiva de luz de backlight	-
Led-	20	LCD_BKL	Polarización negativa de luz de backlight	RC2

Tabla 5.1 Detalle de conexión de puerto del display GLCD

5.3.2 Configuración del puerto serial

Figura 5.5 Circuito de acoplamiento RS-232 / TTL



El puerto serial es implementado para la recepción de datos desde el GPS ó para comunicación con una computadora personal, utilizando un conector DB9. La etapa de acoplamiento se realiza a través del circuito integrado MAX232 que tiene dos pares de convertidores de nivel lógico RS-232

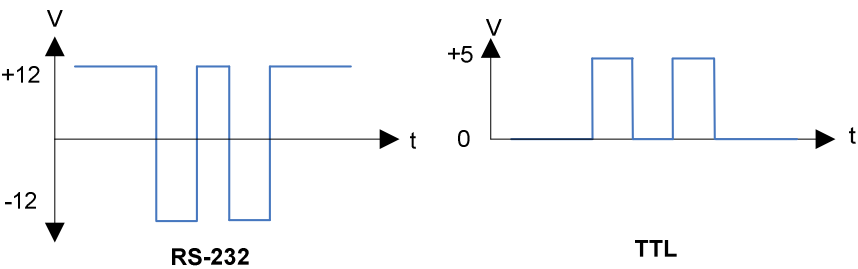
y TTL/CMOS, mas una polarización de 5V que los toma del sistema. El circuito integrado posee internamente circuitería de conversión DC / DC para excitar las salidas RS-232 con el voltaje y corriente estándar. La configuración de pines del conector DB-9 y el circuito se muestran a continuación:

Líneas desde el conector DB-9	#PIN	Nombre	Descripción	Líneas hacia el microcontrolador
NC	1	Not Connected	No conectado	-
Tx	2	Transmit Data	Transmisión de datos hacia el PC	RA5
Rx	3	Receive Data	Recepción de datos desde el PC ó GPS	RA4
NC	4	Not Connected	No conectado	-
GND	5	Ground	Tierra	-
NC	6	Not Connected	No conectado	-
NC	7	Not Connected	No conectado	-
NC	8	Not Connected	No conectado	-
NC	9	Not Connected	No conectado	-

Tabla 5.2 Detalle de conexión del puerto serial RS-232

Los niveles de señal que poseen tanto el microcontrolador como el dispositivo RS-232, es opuesto ó inverso. Los conversores DC/DC internos del circuito de acoplamiento realizan la inversión de niveles mediante un negador lógico (ver siguiente figura).

Figura 5.6 Comparación de señales en el circuito de acoplamiento RS-232 / TTL



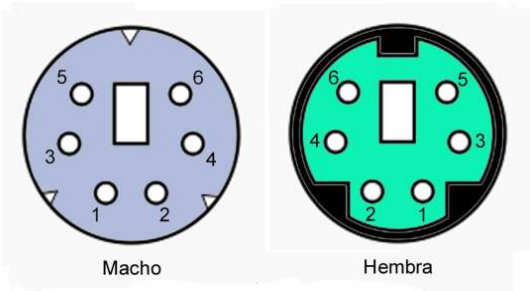
5.3.3 Conexión GPS por puerto serial

El receptor GPS Globalsat® BR-355 viene encapsulado en un armazón plástica herméticamente sellada y provista de un cable con conector PS/2 para señales de datos y polarización. La configuración de pines del conector PS/2 es distinta al estándar de conexión de dispositivos PS/2 como Mouse y Teclado. La comparación entre la conexión estándar PS/2 y del receptor GPS, y el esquema gráfico de los conectores se detallan a continuación:

#PIN	PC Normalizado	GPS BR-355
1	Data	Gnd
2	NC	Vcc
3	Gnd	NC
4	Vcc	Rx
5	Clk	Tx
6	NC	NC

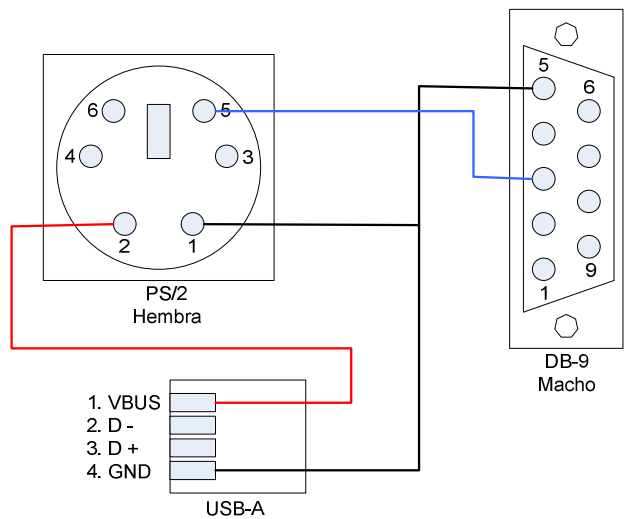
Tabla 5.3 Comparación de conexiones PS/2 entre GPS y PC normalizado

Figura 5.7 Conectores PS/2 macho y hembra



La alimentación para el receptor es 5V, que el conector RS-232 no posee; entonces es necesario la implementación de un adaptador para las señales de datos y para la alimentación por medio de un conector DB-9 y USB-A.

Figura 5.8 Adaptador DB-9/USB-A a GPS PS/2



La configuración del software para comunicación serial con el GPS se implementa con las especificaciones del protocolo NMEA0183 (ver capítulo 2).

5.3.4 Configuración del puerto de la tarjeta SD

Como se observó en el capítulo de marco teórico, la tarjeta de memoria externa Secure Digital puede funcionar en dos modalidades principales: SDIO y SPI. En el presente caso se trabaja con la modalidad de bus SPI (serial peripheral interface) el cual posee cuatro líneas de comunicación:

SD_DI	->	Data Input	->	Hacia el controlador
SD_DO	->	Data output	->	Hacia la tarjeta
SD_CLK	->	Clock	->	Hacia la tarjeta
SD_CS	->	Chip Select	->	Hacia la tarjeta

Para cumplir con las especificaciones de voltaje de entrada y salida de la tarjeta SD, es necesario conocer el rango de trabajo entre VDD y VSS dado por el siguiente gráfico.

Figura 5.9 Gráfico de niveles de voltaje en el bus de la tarjeta SD

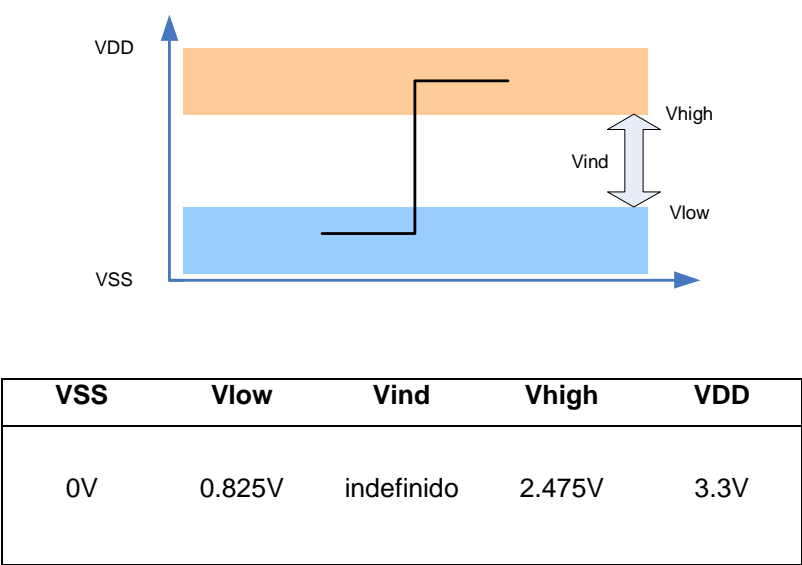
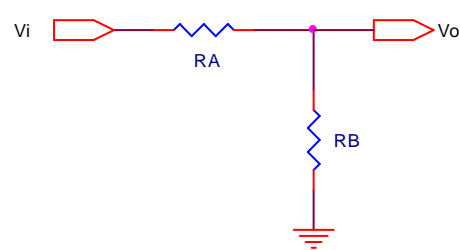


Tabla 5.4 Valores de niveles de voltaje en el bus de la tarjeta SD

El microcontrolador posee señales TTL de 5V y un circuito de acoplamiento es necesario para comunicar las tres líneas que van hacia la tarjeta. A continuación se calcula un divisor de tensión para realizar esta tarea:

Figura 5.10 Circuito de acoplamiento TTL a SD



Sea $RB = 3.3K\Omega$

$$Vo = Vi \cdot \frac{RB}{RA + RB}$$

$$RA = \frac{RB(Vi - Vo)}{Vo} = \frac{3.3K\Omega(5V - 3.3V)}{3.3V}$$

$$RA \approx 1.8k\Omega$$

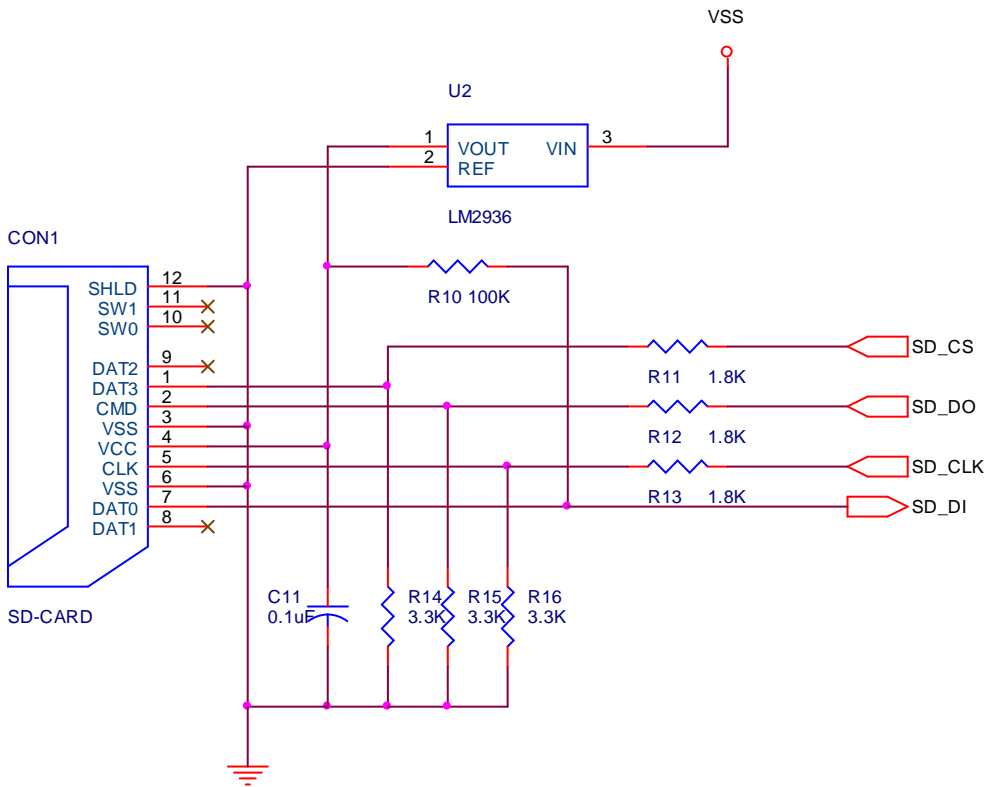
La alimentación de la tarjeta de memoria se la realiza por medio de un regulador de voltaje de 3.3V a través del circuito integrado LM2936 que toma la potencia a partir de la fuente regulada de 5V del sistema. Los valores de consumo de corriente de la tarjeta de memoria se especifican a continuación:

Modalidad	Valor	Unidad	Nota
Sleep	250	uA	Máximo
Lectura	65	mA	Máximo
Escritura	75	mA	Máximo
VDD : 2.7V a 3.6V			

Tabla 5.5 Valores de consumo de corriente de la memoria SD

A continuación se presenta el detalle de conexiones del zócalo receptor SD y hacia las líneas del microcontrolador para modalidad SPI, y la configuración del circuito.

Figura 5.11 Diagrama de conexión del zócalo de tarjeta SD



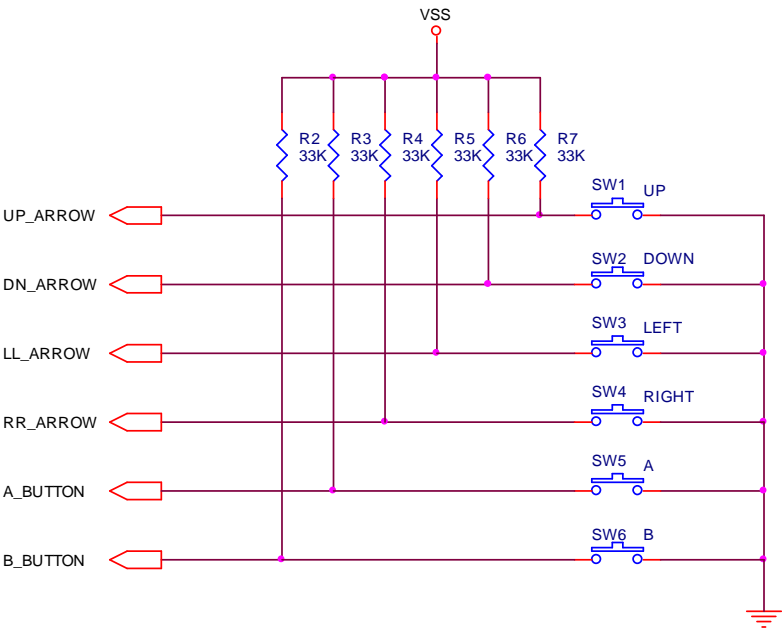
Líneas Zócalo SD	Nombre	#PIN	Descripción	Líneas hacia el microcontrolador
DAT3	SD_CS	1	Chip Select	RE0
CMD	SD_DI	2	Data In	RB0
VSS	GND	3	Tierra	-
VCC	+3,3V	4	+3,3V	-
CLK	SD_CLK	5	Reloj	RB1
VSS	GND	6	Tierra	-
DAT0	SD_DO	7	Data Out	RC7
DAT1	NC	8	No Conectado	-
DAT2	NC	9	No Conectado	-
SW0	NC	10	No Conectado	-
SW1	NC	11	No Conectado	-
SHLD	GND	12	Tierra Carcaza Metálica	-

Tabla 5.6 Detalle de conexión de puerto de la tarjeta SD

5.3.5 Configuración del puerto de teclado

El teclado consta de dos grupos de botones que forman el control de cursor (flechas superior, inferior, izquierda y derecha), y las teclas función de menú y display (A y B). La interfaz es paralela con resistores de pull-up en cada línea.

Figura 5.12 Circuito de interfaz de teclado



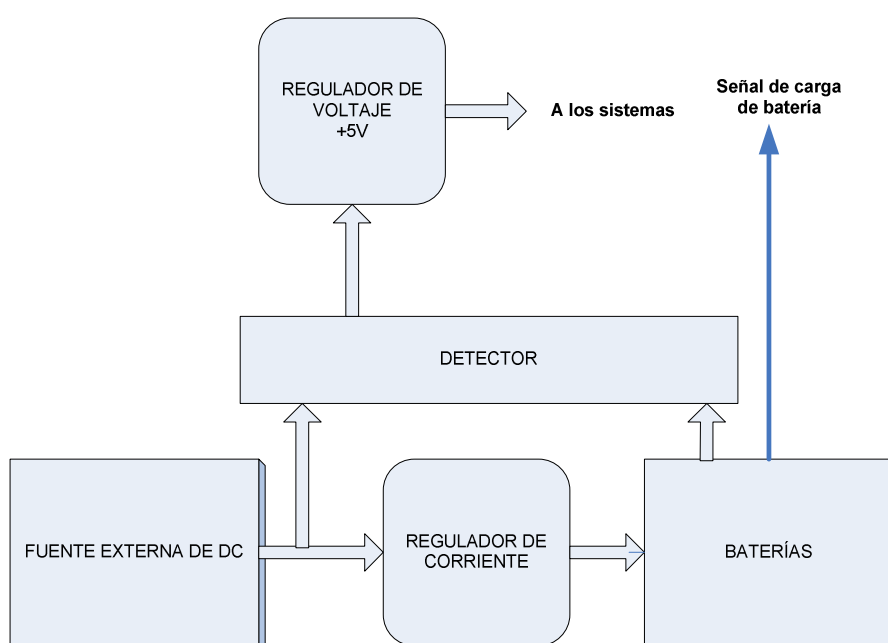
Líneas desde el teclado	#PIN	Descripción	Líneas hacia el microcontrolador
UP_ARROW	1	Cursor Flecha Arriba	RC4
DN_ARROW	2	Cursor Flecha Abajo	RC0
LL_ARROW	3	Cursor Flecha Izquierda	RC1
RR_ARROW	4	Cursor Flecha Derecha	RC5
A_BUTTON	5	Botón A	RC6
B_BUTTON	6	Botón B	RE3

Tabla 5.7 Detalle de conexión de puerto de teclado

5.3.6 Fuente de poder

La fuente de poder es el circuito que se encarga de administrar valores de voltaje y corriente correctos para el correcto funcionamiento del dispositivo a través de baterías o de adaptador externo. La fuente se forma por dos bloques principales: un bloque regulador de voltaje que proporciona 5 voltios DC para alimentar el microcontrolador y todos los periféricos, y otro bloque regulador de corriente que se encarga de suministrar el valor de corriente correcta para la carga de las baterías.

Figura 5.13 Diagrama en bloque de la fuente de poder



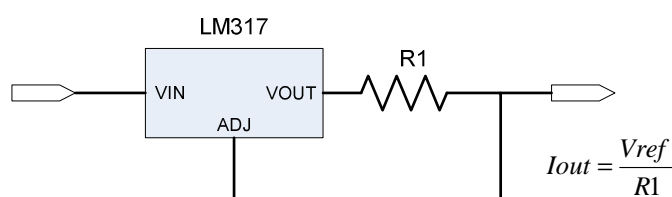
El nivel de carga de batería se supervisa a través de software por medio de señal enviada por el conversor A/D del microcontrolador. En el siguiente gráfico se muestra las etapas simplificadas de la fuente de poder (ver figura 5.13).

Los reguladores de corriente y voltaje están constituidos por los circuitos integrados LM317 y LM7805 respectivamente que pueden manejar corrientes de hasta 1A, que en el presente caso es suficiente pues en la práctica no se alcanzará dicho nivel de corriente, ni en casos extremos.

5.3.7 Circuito regulador de corriente constante

La regulación de corriente está presente en el momento que se utilice un adaptador externo y la recarga de baterías sea inicializada. Un valor de corriente constante determinado por el fabricante de baterías debe ser suministrado para la recarga. En el presente caso se utilizará una batería de 8.4V de Ni-MH de 170mA/h, cuya corriente de recarga nominal es de 17mA en 16 horas. El diseño del circuito se detalla a continuación:

Figura 5.14 LM317 como fuente de corriente constante



$$I_{out} = \frac{V_{ref}}{R1}$$

$$R1 = \frac{V_{ref}}{I_{out}} = \frac{1.25V}{0.1A} = 12.5\Omega$$

$$Potencia = \frac{V_{in} - V_{ref}}{R1} = \frac{15V - 1.25V}{12.5\Omega}$$

$$Potencia = 0.3W$$

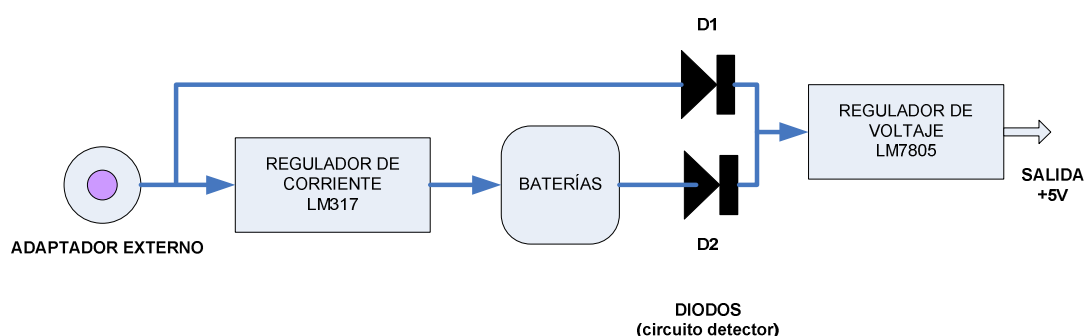
5.3.8 Circuito regulador de voltaje constante

La regulación de voltaje se la realiza por medio del circuito integrado LM7805 que puede manejar hasta 1A, con una variación de voltaje de entrada de 7V a 30V.

5.3.9 Circuito detector

El circuito detector permite seleccionar de manera automática la alimentación desde las baterías o desde la fuente externa hacia el resto del dispositivo. Este circuito permite también habilitar la carga de las baterías mientras exista una corriente externa desde un adaptador. El detalle se muestra a continuación:

Figura 5.15 Diagrama simplificado del circuito detector en la fuente



Inicialmente el circuito regulador de tensión de 5V (LM7805) se encuentra alimentado por la corriente proveniente de las baterías por medio del diodo D2 que en ese momento se encuentra conduciendo. Al no existir conectado un adaptador externo, el diodo D1 se encuentra abierto por ausencia de corriente. Al conectar un adaptador externo con voltaje mayor que el voltaje de las baterías internas (12 voltios en este caso), el diodo D1 se polarizará directamente y empezará a conducir corriente hacia el circuito regulador de 5 voltios. Debido a que en este momento existe un voltaje mayor que el de las baterías en el cátodo de D1 y D2, el diodo D2 deja de conducir ya que no se encuentra polarizado directamente; por lo tanto las baterías entrarán en modo de recarga y no fluirá ninguna corriente de consumo proveniente de éstas.

5.3.10 Circuito de acoplamiento para medición de carga de batería

La señal que la batería debe suministrar al microcontrolador debe estar en el rango manejable de la entrada A/D, es decir la señal debe estar entre los valores máximos de 0 y 5V. Considerando una entrada máxima de voltaje de 15V, para una salida de 5 voltios, tenemos el cálculo del divisor de tensión siguiente:

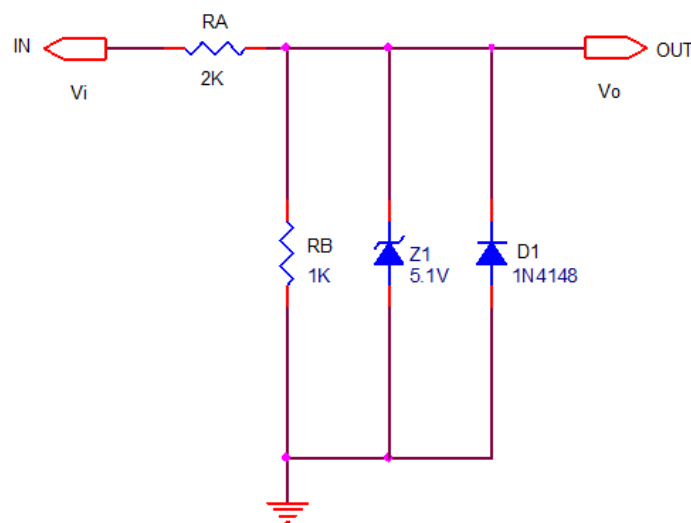
$$V_o = V_i \frac{R_B}{R_A + R_B}$$

$$\text{Sea } R_B = 1k\Omega$$

$$\text{Entonces, } R_A = \frac{R_B \cdot (V_i - V_o)}{V_o}$$

$$R_A = \frac{1k\Omega \cdot (15 - 5)}{5} = 2k\Omega$$

Figura 5.16 Diagrama del circuito de acoplamiento para medición de carga de batería



El diodo Z1 es un diodo zener de 5.1 voltios colocado en polarización inversa; el cual protege la entrada del microcontrolador de un posible sobrevoltaje ante la falla de alguna etapa anterior. El diodo D1 es un diodo común configurado inversamente que protege ante una posible entrada de voltaje negativo hacia el microcontrolador, esta situación puede darse al tratar de conectar erróneamente la batería de 9 voltios.

CAPÍTULO 6

Implementación y Pruebas

C O N T E N I D O

-
- 6.1 Implementación del hardware**
 - 6.2 Generación de archivos de prueba**
 - 6.3 Almacenamiento de archivos en medio físico**
 - 6.4 Pruebas finales con la unidad GNOM-4550**
-

6.1 IMPLEMENTACIÓN DEL HARDWARE

6.1.1 Diseño de placa de circuito impreso

Para el diseño PCB se utiliza el paquete de software *Eagle Layout Editor* de la empresa *Cadsoft®*. Para la simulación de la placa de circuito impreso en 3D se utiliza *Eagle-3D* y *PovRay*; todos bajo el entorno Windows.

A pesar de que *Eagle* posee herramientas de auto trazado a partir del diseño esquemático, se decidió realizar el trazado a mano, debido a que para lograr un diseño portable y pequeño el espacio es importante, y la “destreza” del ruteador automático no es tan fina.

Existen dos placas en el diseño: La placa principal o *MainBoard* y la placa de teclado. Todos los elementos de la circuitería implementada residen sobre la placa principal (*MainBoard*) de doble lado.

Los elementos electrónicos que no existen en librería de *Eagle* se diseñan en el editor de componentes incluido, tales como: pantalla gráfica LCD, zócalo de memoria flash SD, conectores especiales, teclas de cursor y función y sus receptores de presión impresos.

En la simulación 3D de las placas de circuito impreso se generan aquellos componentes que existan en la librería de *Eagle 3D*, mientras que los componentes especiales se muestran solamente como impresión de su silueta sobre la placa.

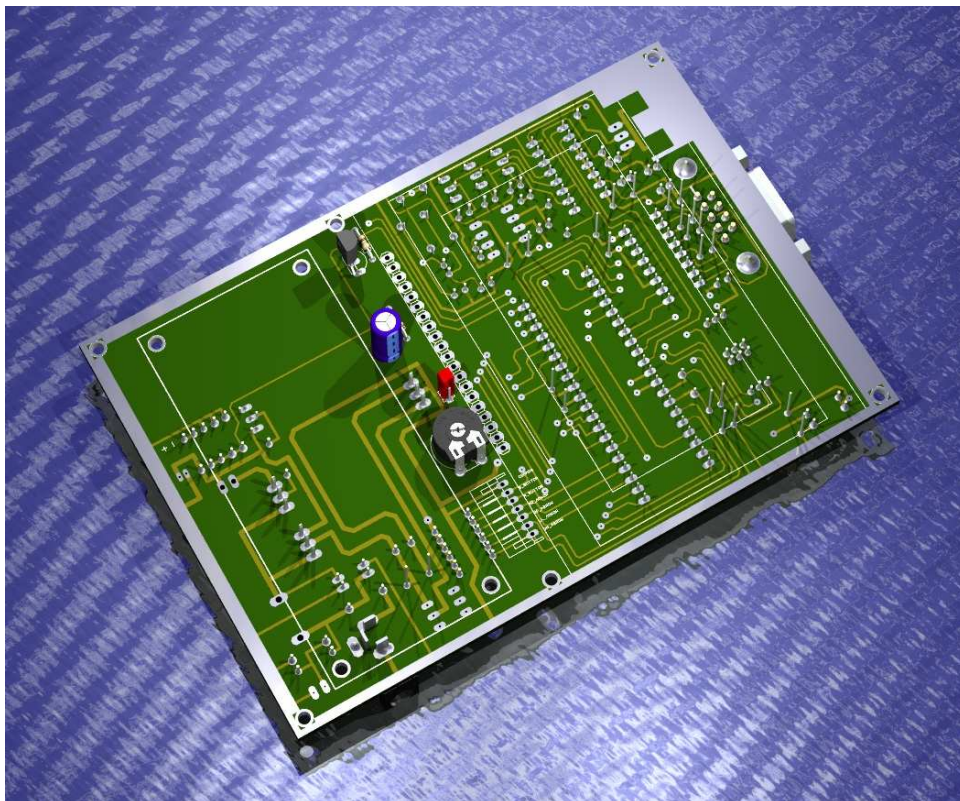


Figura 6.1. Simulación 3D de la placa principal en capa superior (TOP LAYER)

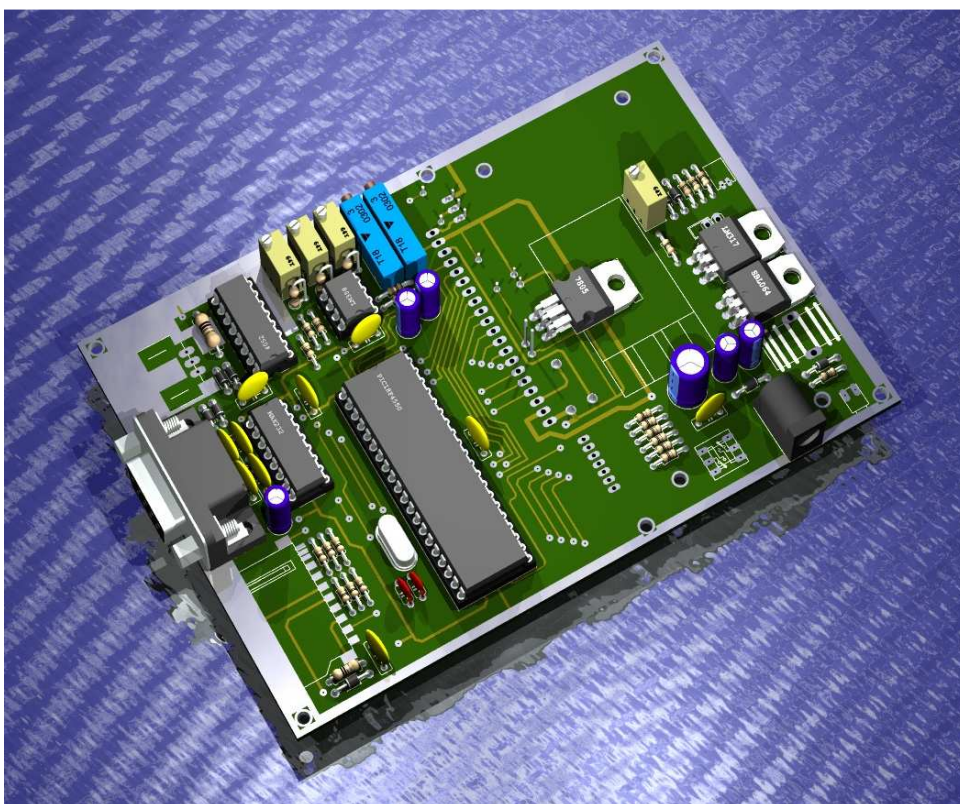


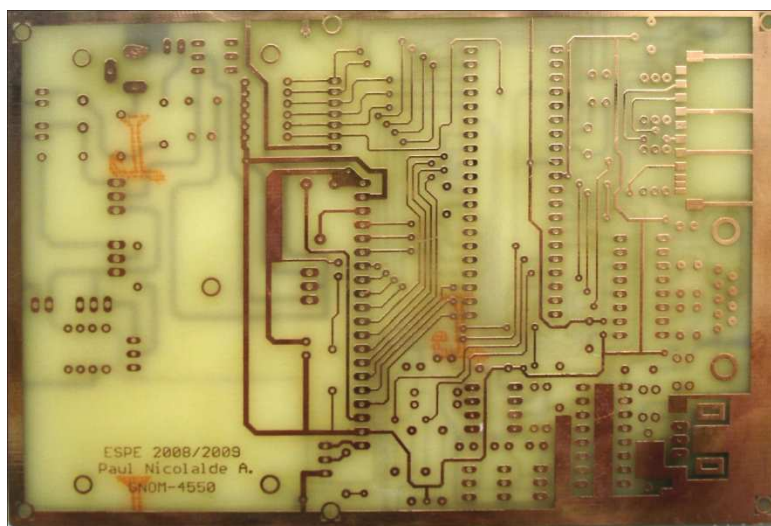
Figura 6.2. Simulación 3D de la placa principal en capa inferior (BOTTOM LAYER)

6.1.2 Preparación de las placas de circuito impreso

En el presente diseño se utilizó la técnica de fotograbado para realizar las placas PCB. Esta técnica permite tener una impresión con buena resolución sobre la placa a doble cara y es un método muy asequible.

La preparación de las PCB consiste en retirar delicadamente el material de fotograbado de las caras de la placa con una escobilla suave por medio de una herramienta rotatoria con el fin de no malograr las pistas. A continuación se cortan los bordes de las placas conforme el diseño (está implícito que el momento del grabado se utiliza una baquelita ligeramente de mayor área). Se procede entonces a la perforación con brocas micrométricas con tamaño de *drill* de acuerdo al diseño.

Figura 6.3. Cara superior de la placa madre obtenida por fotograbado

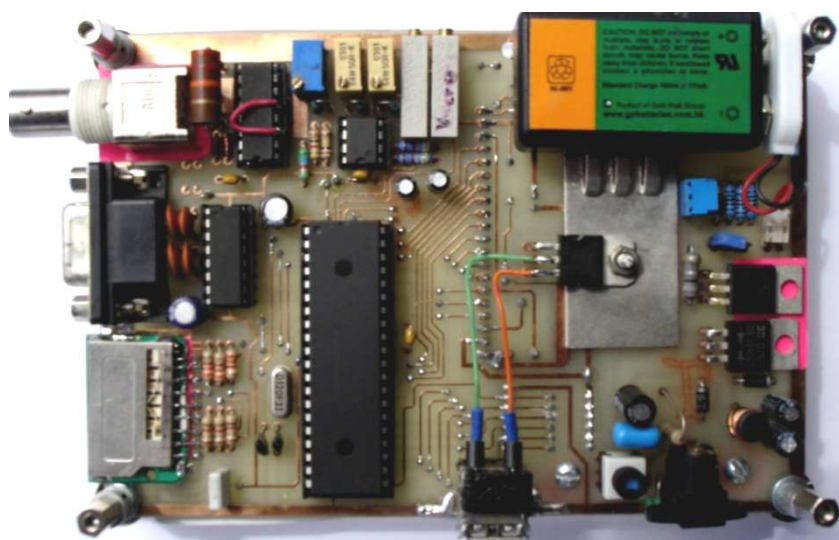


6.1.3 Montaje de los componentes

Con las placas de circuito impreso preparadas, se realiza el montaje de los pasadores (que comunican las señales eléctricas entre capas), previo al montaje los componentes.

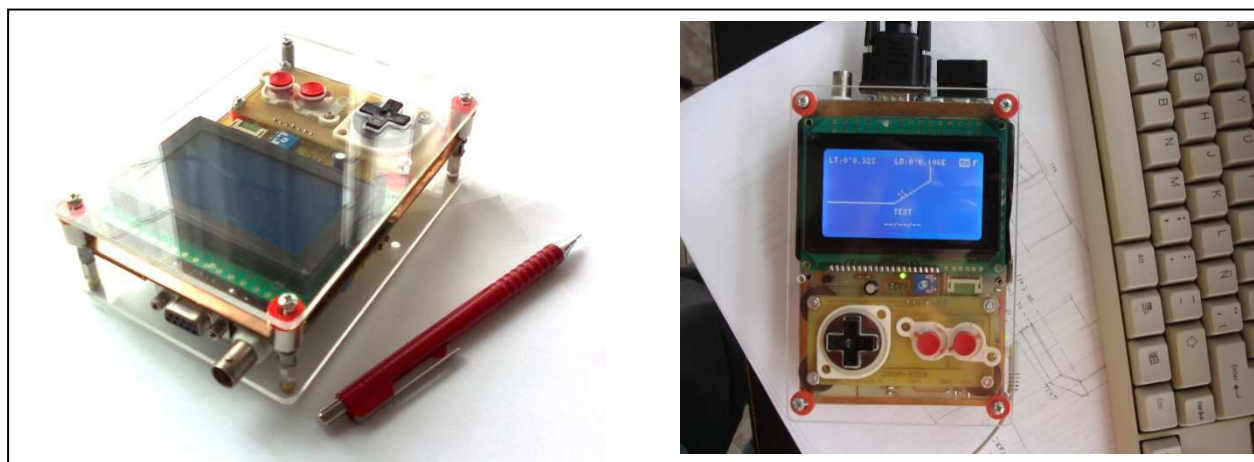
A continuación se procede a soldar los zócalos de los integrados seguidos por resistencias, capacitores, conectores y demás elementos. Por ser un prototipo se requiere montar primero la circuitería de fuente para posteriormente comprobar la correcta polarización de los circuitos integrados y módulo LCD antes de montar el resto de componentes.

Figura 6.4. Vista inferior de la placa madre con componentes montados

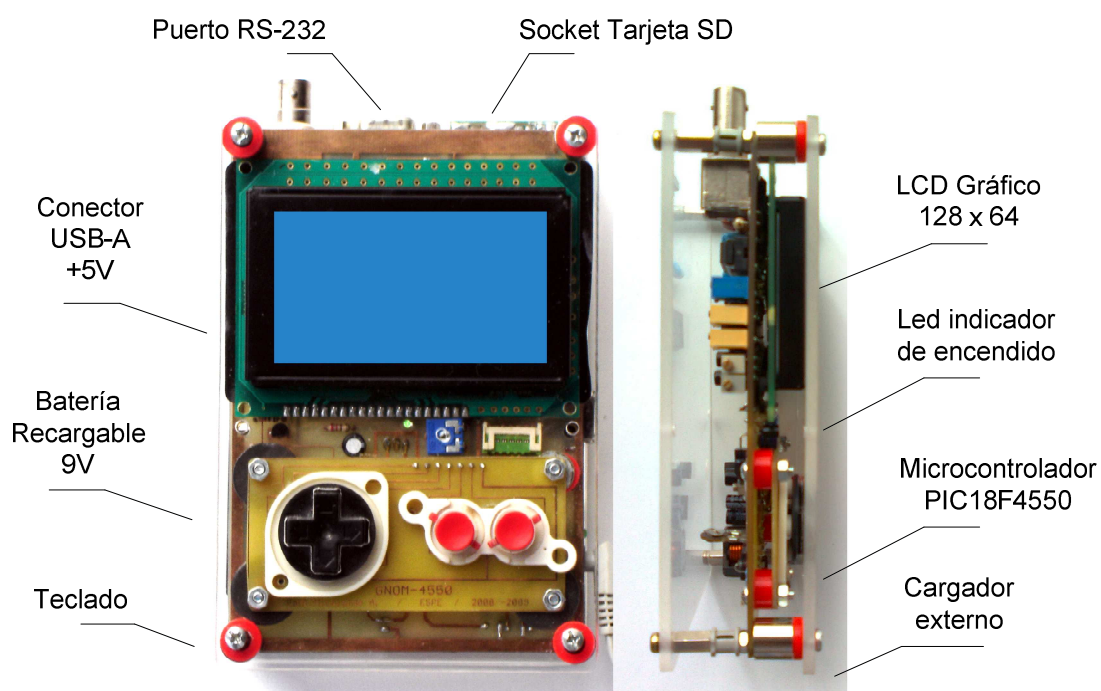


6.1.4 Ensamblaje sobre material protector

El prototipo se ensambló entre dos láminas de material acrílico transparente que provee cierta protección para el manejo del circuito. Las láminas acrílicas también sujetan a los botones del teclado para que se mantengan fijos con respecto a sus respectivos electrodos impresos en la placa.

Figura 6.5. Ilustración del prototipo ensamblado

Las dimensiones del prototipo ensamblado dentro de sus láminas protectoras son 10cm de ancho por 15cm de alto y 4cm de espesor.

Figura 6.6. Presentación del prototipo GNOM-4550

6.1.5 Detalle de costos de fabricación

A continuación se presenta una tabla del costo directo que representa la adquisición de los elementos para el ensamblaje del prototipo al precio del mercado nacional.

COSTO DIRECTO DE MATERIALES	
DESCRIPCIÓN	PRECIO (USD)
Receptor GPS serial GlobalSat BR-355	65
Pantalla LCD de gráficos 128 x 84	38
Microcontrolador PIC18F4550	10
Tarjeta de memoria flash SD de 1GB	22
Placa de circuito impreso fotograbada	11
Zócalo de memoria SD	10
Elementos activos varios	8
Elementos pasivos varios	5
Material de protección externa	3
Cables, conectores, borneras, etc.	8
TOTAL	180

Tabla 6.1. Costo directo de implementación del prototipo

6.2 GENERACIÓN DE ARCHIVOS DE PRUEBA

6.2.1 Consideraciones iniciales

Con el fin de realizar las pruebas de la unidad GNOM-4550, es necesario generar un grupo de archivos de navegación tipo “GMV”, de un área geográfica de interés para constatar el funcionamiento del prototipo como navegador GPS en dicha área.

Es necesario llevar el navegador GPS hacia áreas que representen intersecciones entre archivos que constituyen puntos críticos donde la unidad GNOM-4550 ocupa la mayor cantidad de recursos de procesamiento y lectura de datos desde la memoria SD. Para realizar pruebas en por lo menos una de estas intersecciones, es necesario generar 4 archivos GMV contiguos; uno al Noreste, otro al Noroeste, otro al Sureste y un último al Suroeste.

Es necesario que los elementos gráficos descritos en cada archivo contengan líneas ó figuras geométricas consecutivas entre un archivo y otro, para comprobar la continuidad del trazo en las intersecciones de archivos GMV.

Se debe crear elementos categorizados por capas para comprobar el funcionamiento del generador de gráficos y la discriminación en el trazado por capas.

Debido a las limitaciones de procesamiento del navegador GNOM-4550, es necesario tener en cuenta las características de los archivos GMV que pueden ser interpretados por el prototipo:

- Tamaño de encabezado: 1 bloque (byte H=1)
- Número de capas por zona: 3 (byte L=3)
- Tamaño de puntero de memoria: 16 bits / 2 bytes (byte P=2)
- Tamaño de zona mínima: 1 bloque por capa
- Tamaño de zona máxima: 59 bloques (incluyendo las 3 capas)
- Tamaño mínimo de archivo: 25 Kbytes
- Tamaño máximo de archivo: 33.5 MBytes
- La información de zonas adyacentes no debe compartir un mismo bloque

Al iniciar el navegador GPS en el prototipo GNOM-4550, el sistema inicializa sus registros en la posición geográfica 0 grados 0 minutos y 0 segundos de longitud este, y 0 grados 0 minutos y 0 segundos de latitud sur. Si el navegador no actualiza sus datos hacia una nueva posición mediante el

receptor GPS; dicha posición será la posición inicial, que apunta al archivo S0000E00000.GMV, denominado archivo de inicialización por defecto.

En este punto el navegador buscará en el disco el archivo de inicialización por defecto para abrirlo y mostrar el contenido del mapa digital de la posición media del área que describe el archivo; es decir en el cruce de las zonas 6, 7, 10 y 11.

Existen dos maneras para crear archivos GMV: generarlos byte por byte mediante un editor hexadecimal, y la más fácil; mediante el compilador de archivos GMV que se desarrolló en este proyecto en una versión de prueba, para facilitar la creación de archivos; el cual se abordará mas adelante.

6.2.2 Generación de archivo de inicialización S0000E00000.GMV

El archivo de inicialización tiene un contenido simple solamente de identificación de delimitadores de cada zona para poder visualizar y mover el cursor a través de todo el archivo de inicialización.

Para generar el archivo GMV se utilizará un editor hexadecimal (Winhex en este caso), creando un nuevo archivo con los siguientes parámetros:

- Tamaño de archivo: mínimo posible (25KB)
- Tamaño de encabezado: 1 bloque
- Bloques por capa: 1
- Capas por zona: 3

En el editor hexadecimal, se realiza la creación de un nuevo archivo en la pestaña *File*, y se escoge la opción *Create New File* (ver siguiente figura):

Encabezado – Tabla de punteros de bloque (BP)

Cuerpo del archivo

En el cuerpo del archivo se definen las operaciones gráficas que realizará el sistema del navegador para interpretar los datos en la pantalla LCD. La pantalla de prueba del archivo de inicialización será las 16 zonas del archivo delimitadas por un borde lateral con el texto del nombre de la zona y del archivo en la esquina superior izquierda de cada zona, de la siguiente manera:

Figura 6.10. Gráfica de delimitación de zonas de archivo en inicialización

	ZONE 5 S0000E00000	ZONE 6 S0000E00000

Las operaciones gráficas definidas son:

- Trazo de línea continua con las siguientes coordenadas:

(0,0); (0,249); (249,249); (249,0); (0,0)

Código Hexadecimal:

0xFB, 0x00, 0x00, 0x00, 0xF9, 0xF9, 0xF9, 0x00, 0x00, 0xFF

- Colocar texto con coordenadas (5,5 pixeles): "ZONA x", donde x es el número de zona que corresponda.

Código Hexadecimal:

0xFE, 0x05, 0x05, (código ASCII del texto), . . . , 0xFF

- Colocar texto con coordenadas (5,12 pixeles): "S0000E00000".

Código Hexadecimal:

0xFE, 0x05, 0x0C, (código ASCII del texto), . . . , 0xFF

El código hexadecimal de cada conjunto de operaciones se debe repetir para todas las 16 zonas del archivo, con el respectivo cambio en la nomenclatura. Para establecer la separación entre cada zona y capa es necesario tener en cuenta el número de bloque al cual le corresponde cada dato. Como se definió en el encabezado, se ocupa una cantidad de un bloque por cada capa con tres capas por zona. El encabezado y preámbulo ocupan el primer bloque; entonces tenemos:

Figura 6.11. Separación entre bloques y relación en bytes del archivo de inicialización

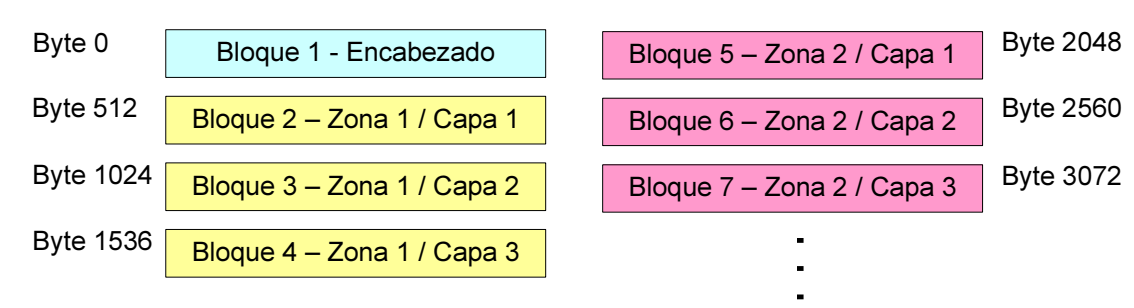
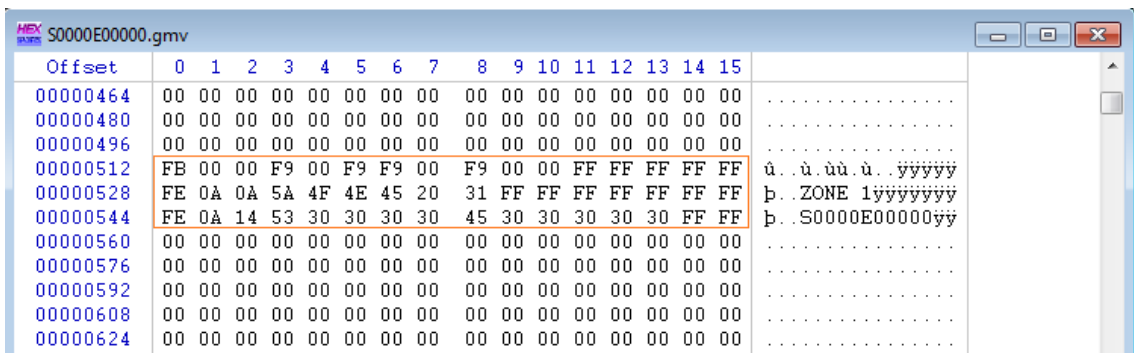


Figura 6.12. Ejemplo de código hexadecimal en bloque de archivo de inicialización



Como se observa en la gráfica anterior, se han descrito tres operaciones gráficas, una por cada línea de 16bytes. Al terminar cada comando, la línea se ha completado con el comando nulo 0xFF. También es posible escribir cada comando seguidamente.

6.2.3 Generación de mapa de la zona de prueba

El área geográfica de interés seleccionada es el campus de la ESPE; en el cual parte de su extensión se encuentra ubicado en una intersección de dos archivos GMV. Será necesario extender la generación del mapa hacia un área que cubra cuatro archivos GMV para poder realizar la prueba de navegación descrita en las consideraciones iniciales del punto 5.2. La información se genera a partir de la herramienta GoogleEarth®, con ayuda de un programa compilador de archivos GMV que fue desarrollado en este proyecto.

Compilador de archivos GMV

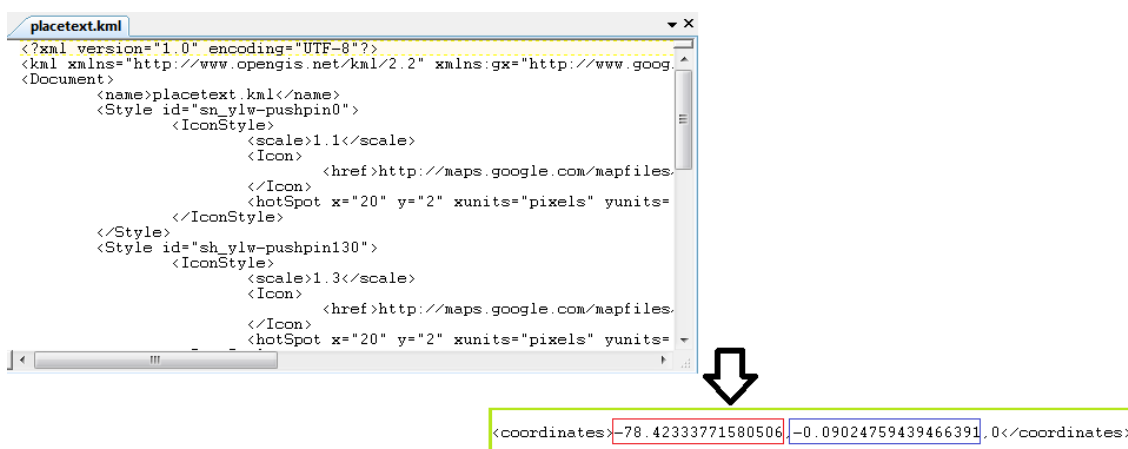
Con el fin de simplificar la tarea de desarrollo de mapas, dentro de este proyecto se ha desarrollado el compilador de mapas GMV en *Visual Basic*; que utiliza archivos generados a partir de *GoogleEarth*, para compilar un mapa en archivos GMV.

KML viene del acrónimo inglés *Keyhole Markup Language*, que es un lenguaje basado en XML para representar datos geográficos en tres dimensiones. Fue desarrollado para ser manejado con *Keyhole LT*, precursor de *GoogleEarth*, luego *Google* adquirió *Keyhole LT* en octubre de 2004.

Un archivo KML contiene entre otra información, una descripción básica de una ruta, punto ó puntos, líneas, polígonos, coordenadas (latitud y longitud), elevación, escalas, nombres, entre otros. El compilador desarrollado en este proyecto extrae únicamente las coordenadas de latitud y longitud; es decir las cadenas de caracteres que representan coordenadas decimales dentro del campo `<coordinates>` (ver figura 6.13).

De la misma manera para que el compilador extraiga y ubique un mensaje de texto, utilizando la herramienta de texto de *GoogleEarth*, la aplicación extrae las coordenadas de posición y el código ASCII del texto del campo `<description>`.

Figura 6.13. Ejemplo de archivo KML en el cual se extraen las coordenadas de puntos de objeto georeferenciado



Construcción de elementos en GoogleEarth

Una vez ubicado el área geográfica de interés en *GoogleEarth*, se procede a dibujar los elementos que contendrá en mapa que queremos crear. Cada ruta, punto ó polígono formado en *GoogleEarth*; constituirá un archivo de salida tipo KML ó un elemento del mapa.

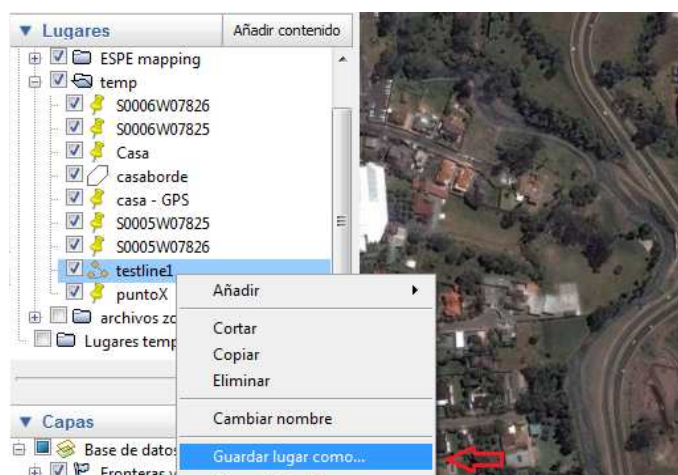
Cabe aclarar que el compilador en su primera versión de prueba (beta), admite archivos KML para utilizar dos herramientas posibles: línea continua y crear texto. Para que el compilador construya un elemento de línea continua, es necesario que el archivo KML fuente, contenga todas las coordenadas de la línea a construir; esto se logra si utilizamos las herramientas de creación de ruta ó creación de polígono en *GoogleEarth*. Para que el compilador construya un elemento tipo texto es necesario que en *GoogleEarth* se cree un elemento puntual que contenga el par de coordenadas de ubicación y el texto propiamente dicho que debe ser escrito en el campo <description>.

Figura 6.14. Creación de elementos de mapa en GoogleEarth a partir de herramientas de ruta, polígono y punto



Para exportar cada elemento del mapa en un archivo KML, es necesario guardar el elemento y seleccionar la extensión “kml” en *GoogleEarth*.

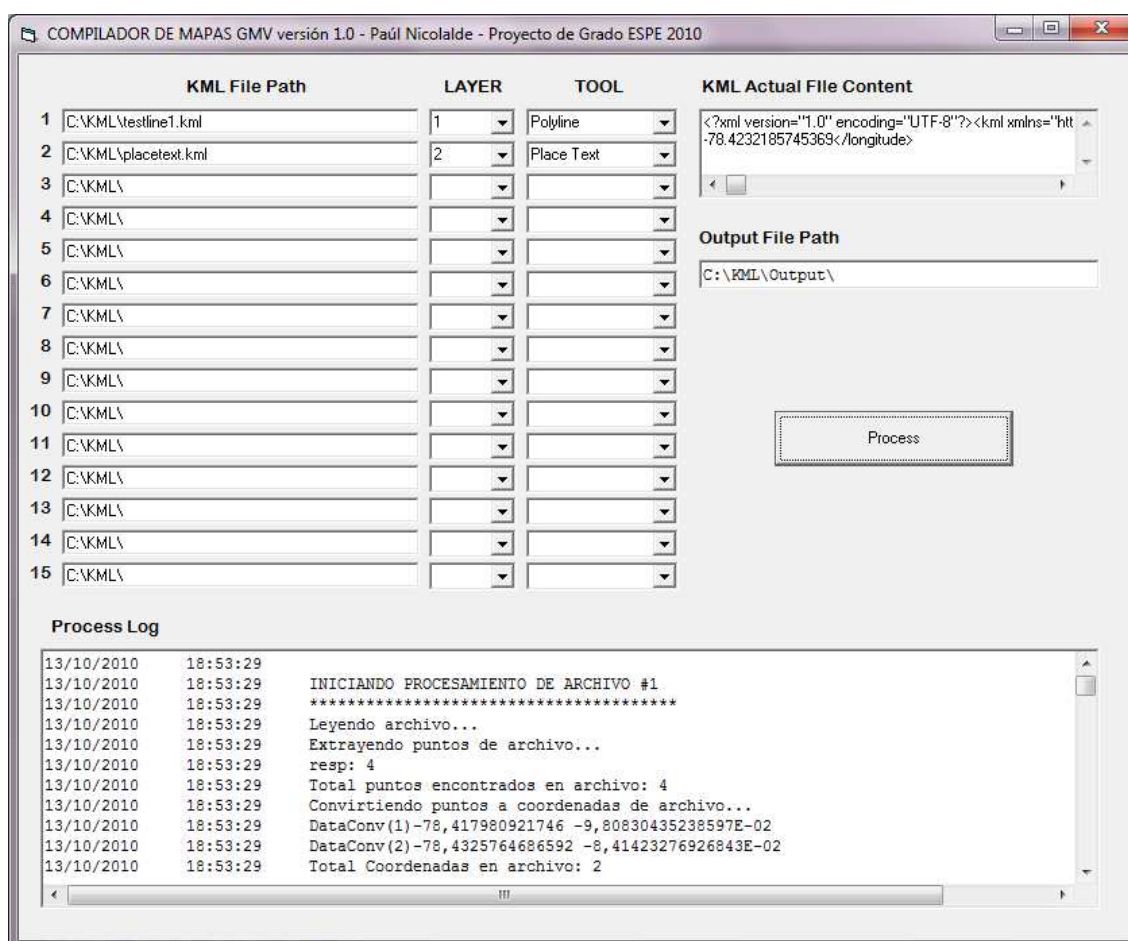
Figura 6.15. Exportación de elementos de mapa como archivos KML en GoogleEarth



Compilación de archivos GMV a partir de archivos KML

En la pantalla del compilador de archivos GMV, se ingresa la ruta y nombre de cada archivo correspondiente a cada elemento del mapa creado en GoogleEarth. El compilador puede recibir hasta quince archivos KML, donde es obligatorio seleccionar la capa a la que pertenecerá cada elemento en el combobox *LAYER*, también se debe escoger el tipo de herramienta con la que el compilador creará el objeto en el mapa (línea continua ó colocar texto) mediante el combobox *TOOL*.

Figura 6.16. Pantalla de programa compilador de archivos GMV

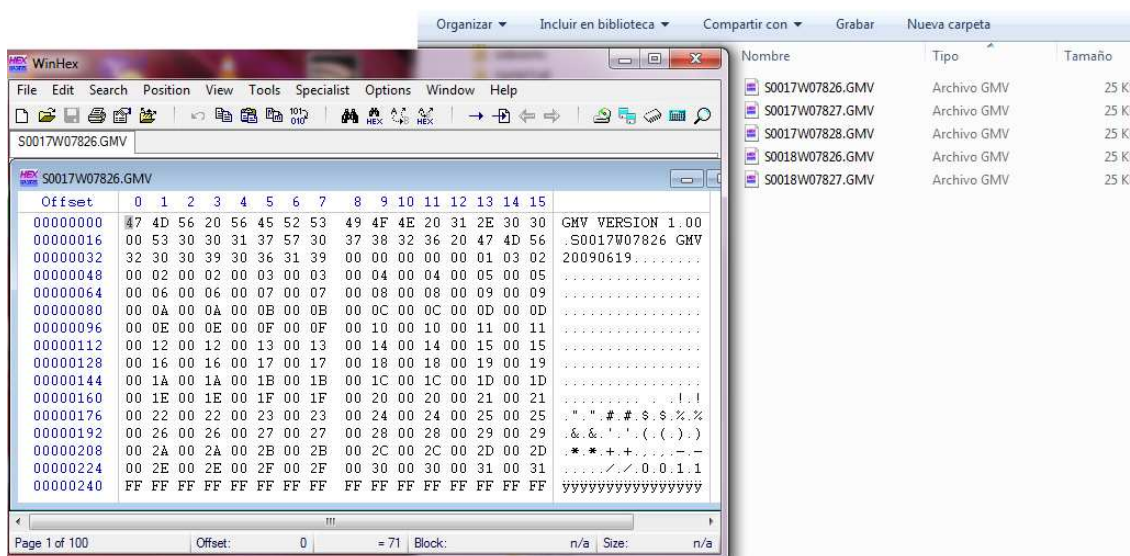


Se debe establecer también la ruta de salida para el conjunto de archivos que van a ser generados en la casilla *OUTPUT FILE PATH*. Por último se inicia la compilación por medio del botón *PROCESS* que mediante la ventana de

texto *PROCESS LOG*, permite observar el detalle del procesamiento que se está realizando.

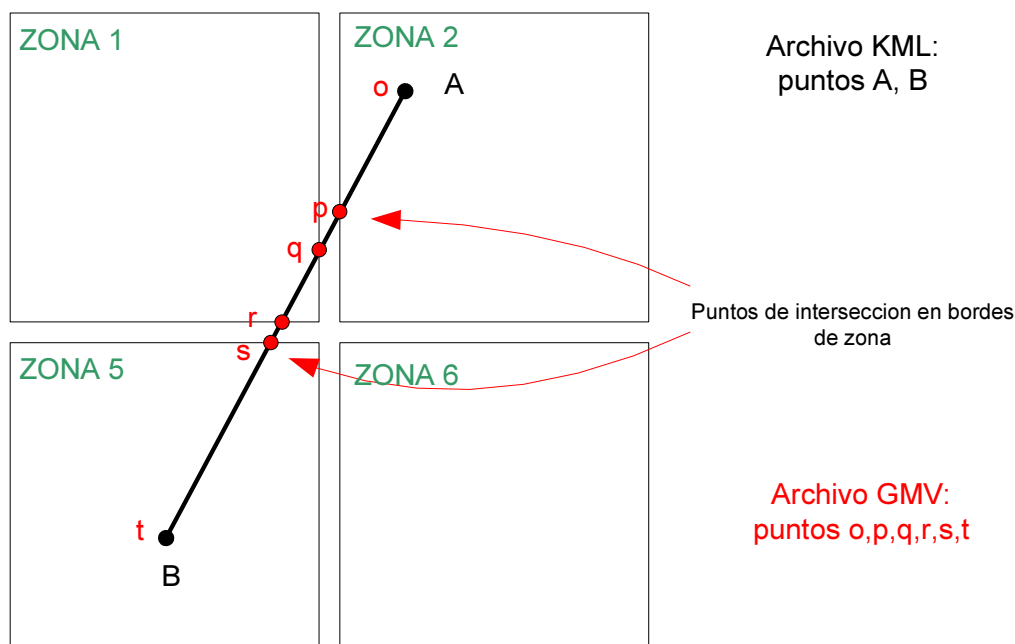
Al finalizar el proceso de compilación y si no se encontró ningún error, se observará en la carpeta de salida todos los archivos que se generaron en la compilación.

Figura 6.17. Archivos GMV de salida del compilador



A pesar de que el aplicativo de compilación de archivos GMV fue desarrollado en el transcurso de este proyecto; no se especifica a profundidad el tema por cuanto no es parte del alcance de este proyecto; sin embargo se describe a breves rasgos el funcionamiento del programa.

Ya que el archivo GMV describe figuras geométricas dentro de un área de 250 x 250 puntos, que corresponde a una zona (ver capítulo 3), el compilador debe poder interpretar la figura geométrica (línea en este caso) contenida en un archivo KML, y calcular todos los puntos que intersecan la línea con los bordes de zona del mapa GMV.

Figura 6.18. Intersección de línea con bordes de zona de un archivo GMV

En el gráfico anterior se observa que cierta línea descrita en un archivo KML con dos puntos (A y B), ocupa geográficamente un área mayor que una zona; por lo tanto el compilador toma esta línea, calcula su ecuación lineal y calcula los puntos de intersección con cada borde de zona (puntos o, p, q, r, s, t), con el fin de que exista la misma figura geométrica en el entorno de un archivo GMV.

En cuanto a la herramienta de colocación de texto, el compilador interpreta la coordenada del punto donde se ubicará el texto y lo traslada a información que soporte el archivo GMV; esto es número de zona y coordenadas dentro de la misma.

El programa compilador realiza este proceso para cada línea y texto que describa en el archivo KML origen, y con esta información y las reglas de estructuración del archivo GMV definidas, construye los archivos binarios de salida necesarios para que el prototipo navegador pueda dibujar en su pantalla la misma información geo-referenciada.

6.3 ALMACENAMIENTO DE ARCHIVOS EN MEDIO FÍSICO

Una vez generados los archivos de navegación, es necesario cargar dichos archivos hacia la memoria SD que utilizará el prototipo para leer el mapa digital de la zona de prueba.

Debido a que se utiliza un sistema de archivos nuevo e incompatible con FAT, se debe realizar el traspaso de los mismos desde una PC hacia el prototipo a través de protocolo Xmodem y cable serial. Para esto se siguen los siguientes pasos:

Paso 1.- Verificar la fuente de alimentación de la unidad; esto es: baterías ó adaptador externo.

Paso 2.- Encender la unidad e introducir la memoria SD en el socket correspondiente (ver siguiente figura).

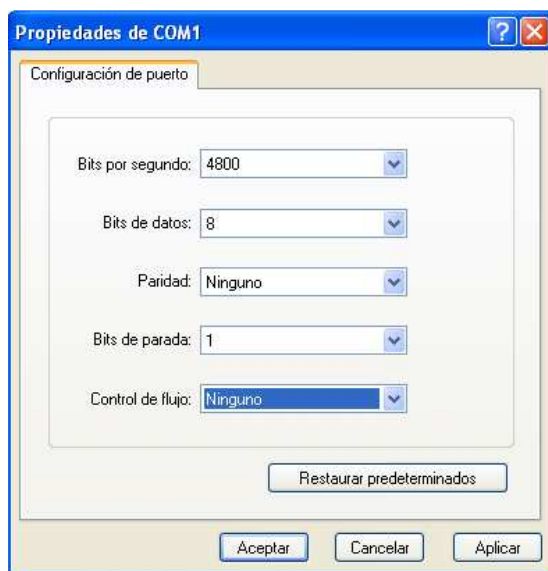
Figura 6.19. Memoria SD en sócalo de la unidad GNOM-4550



Paso 3.- Conectar un cable de transmisión serial DB-9 a DB-9 desde el puerto serial de la PC hacia el puerto serial en el prototipo.

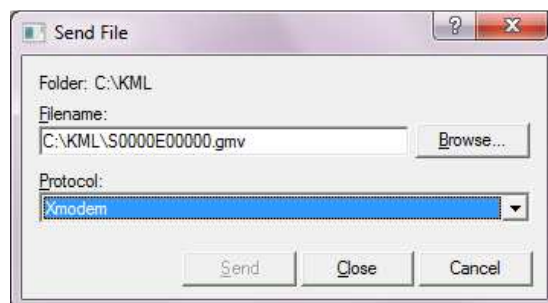
Paso 4.- Abrir el *Hyperterminal* de Windows y configurar un conexión nueva por medio del comunicación serial con características: 4800baud, 8bits de datos, sin paridad, 1 bit de parada, y sin control de flujo (ver siguiente figura).

Figura 6.20. Pantalla de configuración de conexión nueva en Hyperterminal



Paso 5.- Escoger la opción “enviar un archivo” en el menú de “transferir”, con la opción de protocolo Xmodem (ver figura), y seleccionando el archivo GMV a transferir.

Figura 6.21. Pantalla de envío de archivo en Hyperterminal



Paso 6.- Presionar “enviar”. Inmediatamente después, en la unidad escoger la opción “receive file” del menú principal. En este momento el navegador tratará de establecer una conexión con la PC y empezará entonces el envío de datos desde la PC hacia el prototipo, indicando mediante una ventana de aviso el número de paquetes Xmodem recibidos. Este número de paquetes tiene correspondencia con el número indicado en la pantalla de transmisión del *Hyperterminal*.

Figura 6.22. Pantalla de recepción de paquetes en el prototipo



Paso 7.- Un vez finalizada la transmisión, la unidad GNOM-4550 preguntará si se desea guardar el archivo, a lo cual se debe responder que sí, para registrar el archivo recibido en el disco SD.

Para realizar la comprobación de traspaso de archivos hacia la memoria SD, se puede ir a la opción de explorador en el menú principal, y observar el nombre del archivo que se ha enviado; como se observa en la siguiente figura:

Figura 6.23. Pantalla de explorador en prototipo con el archivo recibido



6.4 PRUEBAS FINALES DE LA UNIDAD GNOM-4550

6.4.1 Introducción

Las pruebas se las a dividido en dos. La primera consiste en ejecutar las pruebas de archivo de inicialización, por medio del archivo S0000E00000.GMV, sin señal del receptor GPS. La segunda prueba es la prueba de campo en sí, con el uso del receptor GPS conectado al prototipo; donde se utilizarán los archivos GMV del mapa de la zona de interés, generados previamente. Todos estos archivos deben estar cargados en la memoria SD antes de iniciar las pruebas.

6.4.2 Primera prueba: archivo de inicialización

Paso 1.- Verificar la fuente de alimentación de la unidad; esto es: baterías ó adaptador externo.

Paso 2.- Encender la unidad GNOM-4550 e introducir la memoria SD en el socket correspondiente (ver siguiente figura).

Paso 3.- Escoger la opción “navigate” del menú principal por medio de los cursores del teclado (ver siguiente figura), sin conectar el receptor GPS en el puerto serial, y pulsar el botón de selección B.

Figura 6.24. Menú principal de la unidad GNOM-4550



Paso 4.- Esperar durante el tiempo de inicialización y búsqueda de la señal del receptor GPS.

Figura 6.25. Pantalla de inicialización de receptor GPS



Paso 5.- Al no recibir señal del GPS la unidad preguntará se procede o se reintenta el escaneo. Escoger la opción proceder "Skip" con el botón B.

Figura 6.26. Pantalla de inicialización de receptor GPS (opciones)



Paso 6.- En este momento el navegador carga los registros con la información por defecto que es: 0° 0' 0" longitud Este, y 0° 0' 0" latitud Sur. Inmediatamente el controlador de disco busca el archivo de inicialización S0000E00000.GMV para proceder a leerlo.

Una vez leído el archivo, el cursor se ubica en el centro geográfico del archivo mostrando en pantalla las operaciones gráficas contenidas en él (ver figura).

Figura 6.27. Despliegue de información del archivo de inicialización

Paso 7.- Una vez cargado el archivo de inicialización, se puede navegar por él área en cuestión por medio del cursor del teclado.

Cabe destacar que se observarán las delimitaciones de las zonas del archivo con su respectiva leyenda en la parte superior izquierda de cada zona más el nombre del archivo. Se puede observar también que al mover el cursor fuera de los límites del archivo, el sistema de control genera las delimitaciones por defecto con marcaciones de cada 50 pixeles que corresponden a 50 centésimas de minuto. Si el usuario se aleja por medio de los cursores lo suficientemente del archivo de inicialización, el sistema buscará en el disco los archivos contiguos al S0000E00000.GMV que son los que necesitará para seguir construyendo la gráfica del mapa digital.

6.4.3 Segunda prueba: prueba con receptor GPS

Paso 1.- Verificar la fuente de alimentación de la unidad; esto es: baterías ó adaptador externo.

Paso 2.- Verificar la conexión del receptor GPS en el puerto serial y la alimentación de +5V, en el respectivo conector de alimentación tipo USB – A.

Paso 3.- Encender la unidad GNOM-4550, y ubicar la opción “navigate” en el menú principal (ver figura).

Figura 6.28. Menú principal de la unidad GNOM-4550



Paso 4.- Esperar durante el tiempo de inicialización y búsqueda de la señal del receptor GPS.

Figura 6.29 Pantalla de inicialización de receptor GPS



Paso 5.- Si no se inicializa el receptor GPS, escoger la opción reintentar “retry”, para hasta que la unidad pase automáticamente a la siguiente pantalla. Dependiendo del tipo de receptor GPS, la intensidad de la señal, el registro interno de datos de efeméresis y almanaque; el tiempo que demora en inicializarse el receptor varía desde algunos segundos hasta algunos minutos.

Paso 6.- Luego de haberse inicializado el receptor GPS, la unidad procesa los datos de navegación y busca en el disco los archivos necesarios. Si alguno ó algunos archivos no existen en el disco, la unidad desplegará un mensaje de archivo faltante y procede con el siguiente hasta terminar la actualización en el posicionamiento.

Al desplazarse por el área geográfica, se debe observar el despliegue del mapa en la pantalla de la unidad. Al mover el cursor a partir del teclado, la

unidad pasa al modo manual (MA) donde la actualización de posicionamiento queda deshabilitada para poder mover el mapa a voluntad. Para volver al modo de navegación GPS (GP), hay que pulsar la tecla A y automáticamente la unidad se actualizará con la ubicación del GPS (ver siguiente figura).

Figura 6.30 Pantallas de los modos Manual y GPS de la unidad GNOM-4550

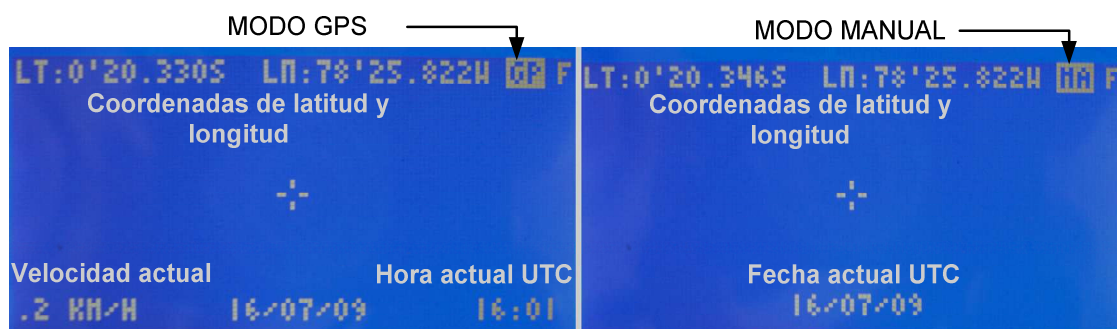
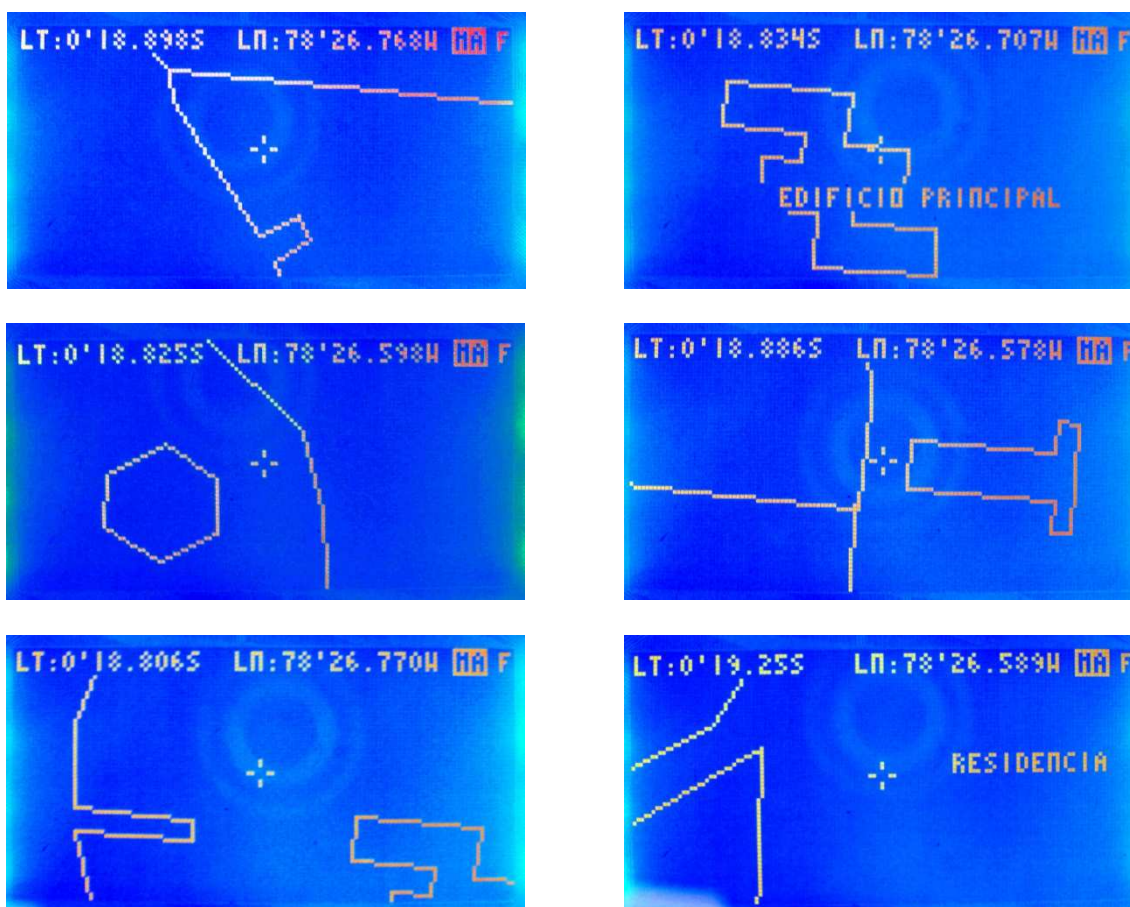


Figura 6.31 Ejemplo de navegación GPS en área de prueba con archivos generados



CONCLUSIONES

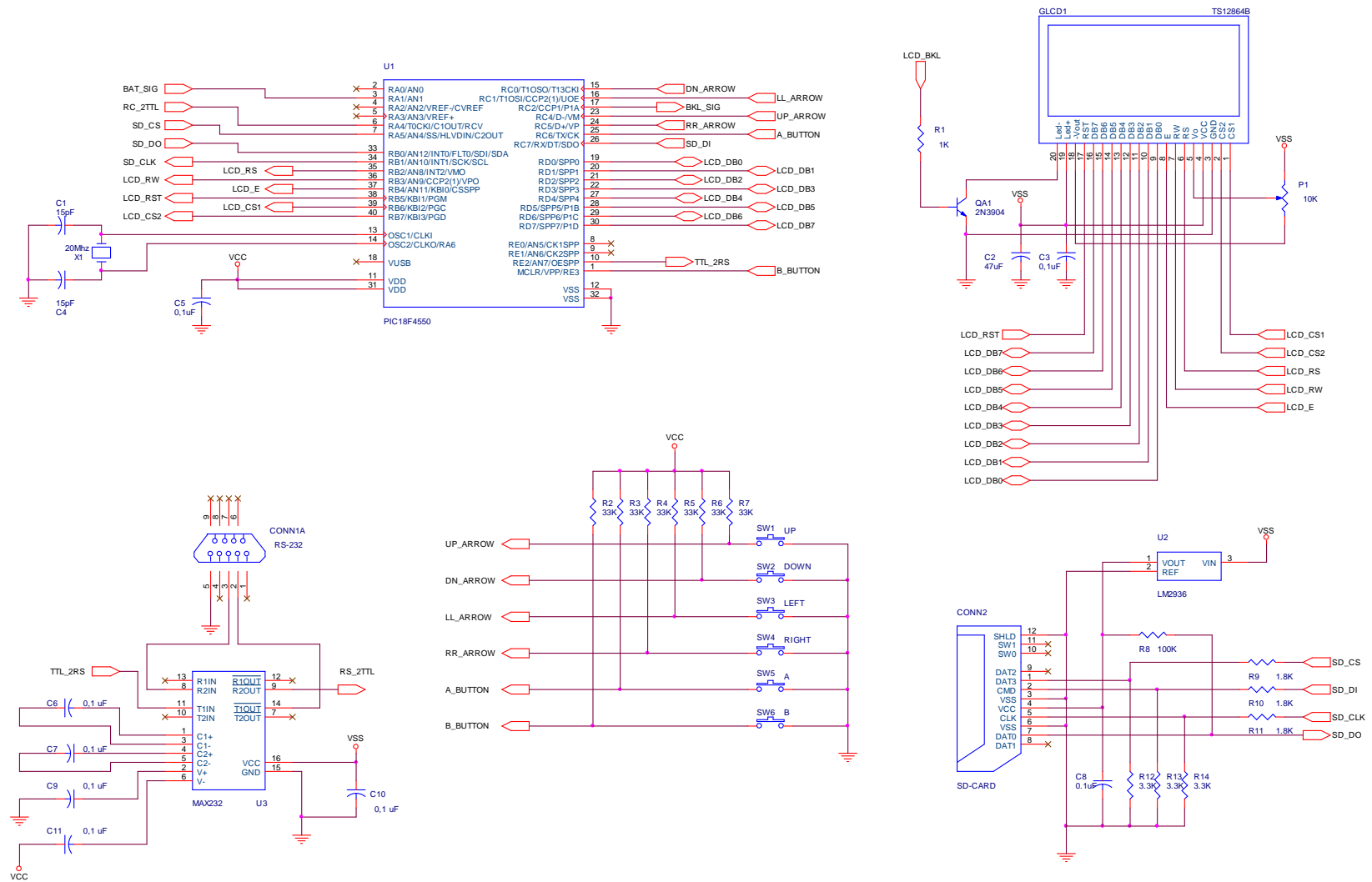
- Mediante el uso de un receptor GPS se logró diseñar e implementar un prototipo electrónico basado en microcontrolador, que permite ser utilizado como un navegador GPS utilizando mapas de áreas geográficas de interés.
- Se ha diseñado un nuevo formato de archivo de mapa digital vectorial geo-referenciado con características que permiten su utilización en sistemas pequeños con microcontrolador.
- En base al nuevo formato de archivo desarrollado, se generó un mapa vectorial de un área geográfica de prueba, con ayuda de un editor hexadecimal desde una PC. Con el mapa contenido en la memoria SD, se realizaron pruebas de la unidad de navegación, las cuales resultaron exitosas cumpliendo con las expectativas planteadas.
- Se ha diseñado también un nuevo sistema de archivos que permite la gestión y almacenamiento de archivos sobre una memoria SD utilizando un microcontrolador.
- La implementación del prototipo GNOM-4550 ha dado como resultado la creación de librerías de control en lenguaje C para el microcontrolador PIC18F4550, en diversas áreas como: recepción de datos con protocolo *Xmodem*, control sobre el sistema de archivos FDZ (creado por el autor), control y transferencia de datos para memorias SD, lectura e interpretación de archivos GMV (creado por el autor).

- Durante el desarrollo del proyecto se implementó un aplicativo desarrollado en Visual Basic que constituye un compilador de mapas GMV básico, que en su versión inicial logra generar archivos de salida compatibles con el prototipo en base de objetos sencillos dibujados dentro del entorno del programa *GoogleEarth*. Esta herramienta facilita la construcción de mapas digitales de prueba para el prototipo desarrollado.

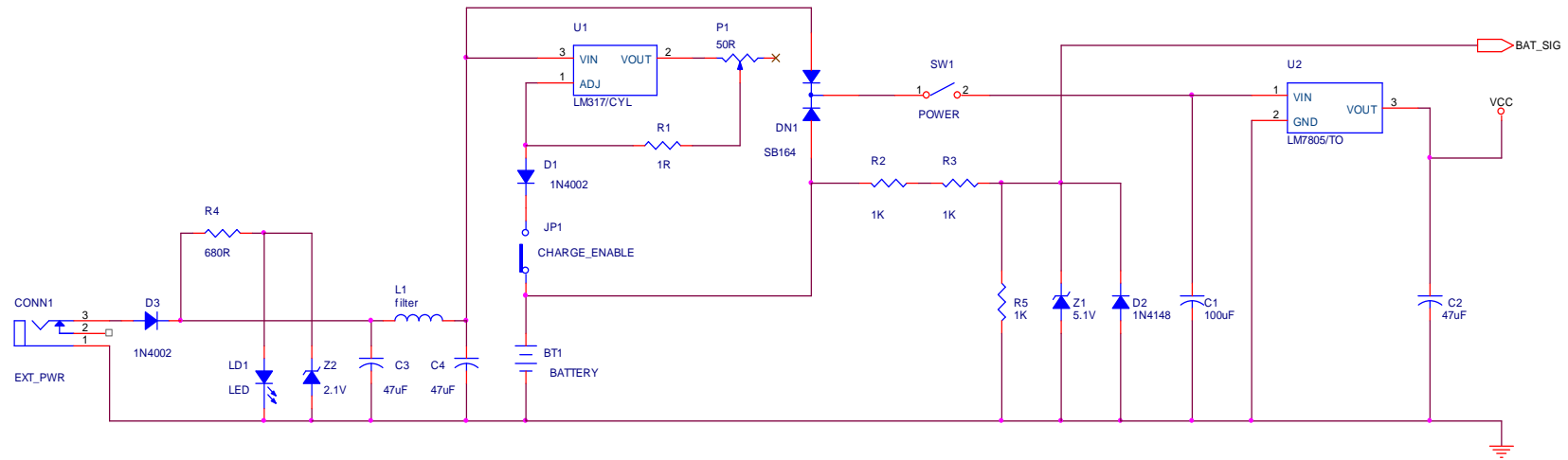
RECOMENDACIONES

- ✓ Una de las recomendaciones para mejorar el desempeño de es utilizar un regulador de voltaje de 5 voltios tipo *SWITCHING*, para disipar la menor energía térmica posible y así alargar la vida de las baterías.
- ✓ Otra opción de mejoramiento de hardware sería implementar la opción de un *BOOTLOADER* que es un pequeño programa residente en el microcontrolador, que mediante el puerto USB y una computadora, permite realizar la carga de cualquier programa a la memoria del microcontrolador sin utilizar un programador externo, ni extraer físicamente el circuito integrado. Por medio de este *BOOTLOADER*, el prototipo se convertiría en una tarjeta de desarrollo para la realización de cualquier aplicación electrónica con una actualización sencilla del programa.
- ✓ El compilador de mapas GMV que fue desarrollado en el transcurso de este proyecto (y que se encuentra fuera del alcance del mismo), constituye una versión de prueba que podría ser mejorada para obtener mayor precisión en el diseño de mapas digitales más complejos.

ANEXO 1
DIAGRAMA ELECTRÓNICO
(Tarjeta principal)

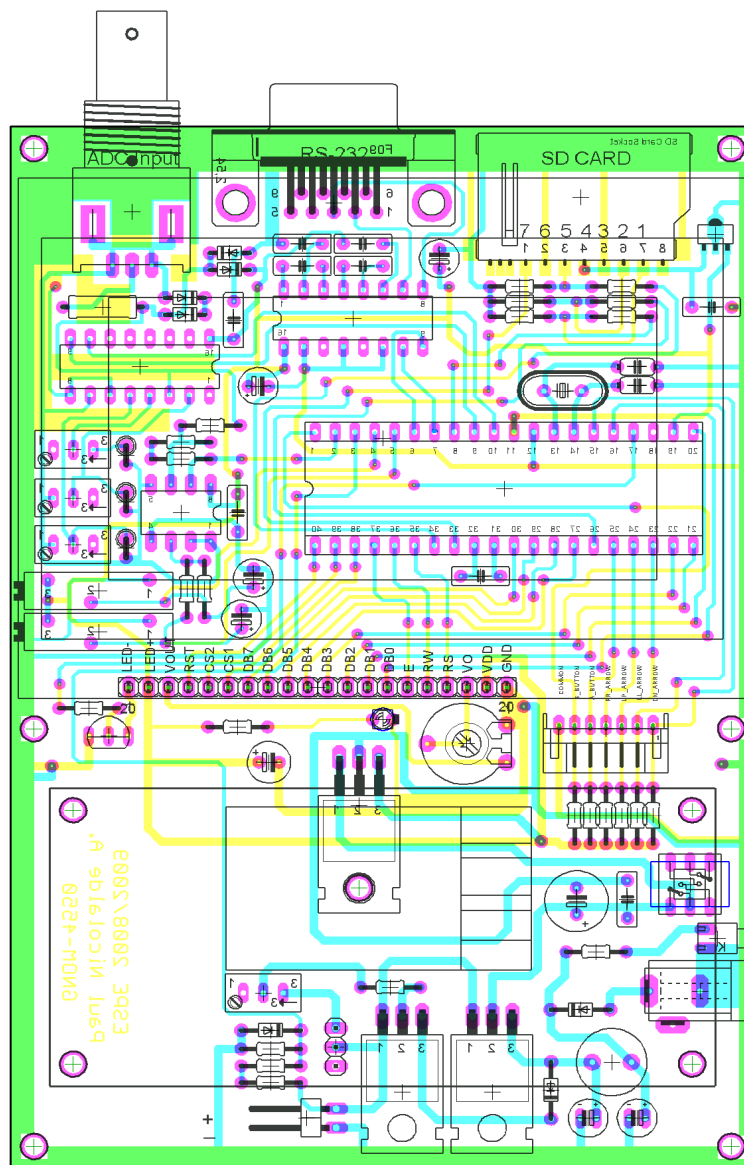


ANEXO 2
DIAGRAMA ELECTRÓNICO
(Fuente de Poder)



Diseñado por: Paúl Nicolalde - proyecto de grado GNOM4550			
Title POWER SUPPLY - GNOM4550			
Size B	Document Number 1		Rev 1A
Date:	Thursday, January 01, 2004	Sheet	1 of 1

ANEXO 3
DISEÑO DE CIRCUITO IMPRESO



NOTA: El diseño actual corresponde al prototipo GNOM-4550 en su versión completa con circuitería de interfaz de entrada A/D mediante conector BNC, preamplificador diferencial con tierra flotante, protecciones contra sobrevoltaje en la entrada analógica.

ANEXO 4

RESUMEN DE PRINCIPALES

FUNCIONES DE LIBRERÍA

SDFDZ16.C (control de tarjeta SD)

```

/*****
//////////////////////////////////SDFDZ16.C//////////////////////////////////
AUTORIA:

    Esta librería fué diseñada completamente por Paul Nicolalde Avila.

DESCRIPCIÓN:

    Librería de implementación de controlador de tarjetas SD
    con de comunicación SPI sobre Sistema de Archivos FDZ16 (Creado por el autor).

Derechos reservados. (C) 2008/2009/2010.

*****/
//////////////////////////////////
///                                COMMAND SUMMARY                                ///
//////////////////////////////////
//=====Initialization Commands=====//
int init_sequence(void);
//=====Format Control=====//
int load_rot_into_ram(void);
int write_rot_into_disk(void);
int calc_fss_and_totalfiles(void);
int erase_fdz(void);
int erase_csi(void);
//=====File and Data Control=====//
int write_unit_element(int16 unit, int index);
int read_unit_element(int16 unit, int index);
int write_data(void);
void xmodem_reset_values(void);
int receive_xmodem_pack(void);
int record_fdz(void);
int make_file(void);
int seek_for_file(void);
//=====Physical Layer Control=====//
void spi_send_word(BYTE dat0,BYTE dat1,BYTE dat2,BYTE dat3,BYTE dat4,BYTE dat5);
int spi_send_byte(BYTE x);
void select_card(int1 mode);
int address_convert(int number);
void dummy_mode(int value);
int check_empty_bus(void);
int read_block(byte add4, byte add3, byte add2, byte add1);
int write_block(byte add4, byte add3, byte add2, byte add1);

```

```

/////////////////////////////////////////////////////////////////
///                                DESCRIPTION                                ///
/////////////////////////////////////////////////////////////////
*****/

int seek_for_file(void){
/* Busca y compara el nombre de archivo de 12 caracteres mas extensión en la
   tabla fdt con el contenido en TRAM. Devuelve 0 si el archivo se encontró y 1
   si no existe. Si el archivo existe coloca la dirección de bloque FDB en las
   variables globales lsbfile y msbfile.
*/
//:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::://
int receive_xmodem_pack(void){
/* Recibe un paquete xmodem de 128 bytes en la memoria XRAM. Envía los caracteres
   de control ACK y NACK e identifica los bytes de control de protocolo. Realiza
   el control de temporización al recibir cada paquete.
*/
//:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::://
void xmodem_reset_values(void){
/* Resetea los valores de las variables utilizadas en la gestión de xmodem para
   una nueva transferencia.
*/
//:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::://
int make_file(void){
/* Al finalizar la recepción de un conjunto de datos por xmodem, cierra el
   archivo completando la información de tamaño de archivo en unidades, y a
   partir de la información contenida en ZRAM crea el bloque FDB correspondiente
   al archivo. Busca un bloque vacío en FDZ y almacena dicho bloque.
*/
//:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::://
int record_fdz(void){
/* Construye el bloque descriptor de archivo en la memoria ZRAM, identificando
   el tipo de archivo. Nombra el archivo conforme a lo recibido. Crea los punteros
   de unidad "UP" en el bloque conforme se almacenan en el disco.
*/
//:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::://
int write_data(void){
/*
   Busca una unidad vacía en la memoria de datos del disco y almacena la
   información de cada bloque de datos contenidos en WRAM, utilizando una unidad
   diferente por cada 58 bloques llenos. Por cada unidad utilizada se contruye
   la tabla de punteros de unidad utilizando la función record_fdz.
*/
//:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::://
int write_unit_element(int16 unit, int index){
/* Escribe un elemento o bloque de datos en una unidad del disco a partir de la
   información contenida en WRAM, con la especificación de número de unidad y
   un índice. El rango de unidad es de 1 a 28912 y el de índice es de 0 a 59.

```

```

*/
//:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::://
int read_unit_element(int16 unit, int index){
/* Lee un elemento o bloque de datos de una unidad del disco a y lo deposita en
   RRAM, con la especificación de número de unidad y un índice.
   El rango de unidad es de 1 a 28912 y el de índice es de 0 a 59.
*/
//:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::://
int erase_fdz(void){
/* Borra el contenido de todos los bloques FDB de la zona de descriptor de
   archivo en el disco. Esta operación equivale a un formateo rápido, pero con
   la utilización del disco se pueden producir errores de cadenas de datos
   perdidas.
*/
//:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::://
int erase_csi(void){
/* Borra el contenido de todos los bloques CSI de la zona de datos del disco.
   Esta operación equivale a un formateo de bajo nivel y tarda varios minutos.
*/
//:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::://
int load_rot_into_ram(void){
/* Lee el primer bloque del disco correspondiente a la tabla ROT y lo deposita
   en la memoria TRAM.
*/
//:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::://
int write_rot_into_disk(void){
/* Durante el formateo del disco esta función escribe la tabla ROT que se
   encuentra en la memoria ROM del procesador hacia el primer bloque de memoria
   correspondiente a dicha tabla.
*/
//:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::://
int calc_fss_and_totalfiles(void){
/* Devuelve el número de archivos totales contenidos en el disco y calcula el
   espacio libre en unidades y lo deposita en la variable fss.
*/
//:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::://
int check_empty_bus(void){
/* Envía ciclos de reloj por el puerto SPI hasta confirmar que el bus de datos
   SPI se encuentre vacío y listo para la siguiente operación.
*/
//:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::://
int write_block(byte add4, byte add3, byte add2, byte add1){
/* Realiza el procedimiento de escritura de bloque en el disco a partir de una
   dirección física de 4 bytes con la información de WRAM. Si la operación es
   exitosa devuelve 0, caso contrario devuelve el valor del error (ver documento).
*/
//:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::://
int read_block(byte add4, byte add3, byte add2, byte add1){
/* Realiza el procedimiento de lectura de bloque del disco a partir de una
   dirección física de 4 bytes depositando la información en RRAM. Si la operación

```

[illegible]

NAVCTRL.C (control de navegación)

```

/*****
/////////////////////////////////NAVCTRL.C/////////////////////////////////
AUTORIA:

    Esta librería fué diseñada completamente por Paul Nicolalde Avila.
DESCRIPCIÓN:

    Librería de control de navegación GNOM-4550.

Derechos reservados (c) 2008/2009/2010.

//:.....//
/////////////////////////////////
///                                DESCRIPTION                                ///
/////////////////////////////////
//:.....//
void init_reference_data(int16 lon_deg,int lon_min,int lon_dir,int lat_deg,int
    lat_min,int lat_dir){
/* Inicializa el registro GPSREG con los datos ingresados por medio de los
    argumentos de la función.
*/
//:.....//
void chase_cursor(void){
/* Realiza el desplazamiento de la posición referencial del núcleo de navegación
    por medio del cambio en los valores del registro GPSREG, en base a la infor-
    mación del cursor NAVREG. (Empata los valores de los registros GPSREG y NAVREG)
    No se realiza lectura de disco en esta operación.
*/
//:.....//
void get_cursor_data(void){
/* Calcula los valores numéricos del cursor en el registro NAVREG en base a la
    posición relativa del núcleo de navegación a través de los valores del registro
    GPSREG, y la posición del vector rc dentro del núcleo.
*/
//:.....//
int vector_position_calc(void){
/* Calcula el vector d1 y d2 a partir del vector cursor (r). Detecta cambios
    de zona para disparar la actualización del registro NAVREG.
*/
//:.....//
void zone_change_detect_and_arm(void){
/* Detecta cuando ocurre un cambio de zona por movimiento del cursor cuando llega
    al área borde del núcleo de navegación para que el sistema actualice los
    registros.

```

```
//:::::::::::::::::::::::::::::::::::::::::::::::::::://
void check_counters(int n){
/* Con cada cambio de zona detectada se modifica la cuenta de los datos
referenciales
*/
//:::::::::::::::::::::::::::::::::::::::::::::::::::://
int search_and_arm(int reference){
/* Busca el/los archivo(s) de navegación necesario(s) en el disco según la
referencia en mención
que existe en el argumento y lo carga en la porción correspondiente en la ram
(ZRAM).
Si no existe el arhivo se informa al programa principal y borra la porción de
memoria
caché (ZRAM) según el argumento.
*/
//:::::::::::::::::::::::::::::::::::::::::::::::::::://
void convert_file_name(int reference){
/* Convierte los datos actuales de navegación a nombres de archivo, basado en la
informaciòn de la memoria de referencia.
*/
//:::::::::::::::::::::::::::::::::::::::::::::::::::://
int read_and_charge_file(int level){
/* Busca en el disco el archivo indicado en tram, interpreta la definición del
archivo y carga la NFDI a la porción de ram (ZRAM) indicada en el argumento
'level'. Los argumentos válidos son 0,1,2 y 3.
*/
//:::::::::::::::::::::::::::::::::::::::::::::::::::://
int refresh_onscreen_map(void){
/* Interpreta la tabla NFDI de la memoria caché ZRAM, extrae la información del
disco como indica la tabla en bloques de datos en RRAM, y dibuja el mapa digi-
tal que contiene cada bloque. Si la tabla NFDI está vacía (archivo no
encontrado),
se dibuja en su lugar los cuadrantes por defecto contenidos en la memoria
ROM nodata[];
*/
//:::::::::::::::::::::::::::::::::::::::::::::::::::://
void draw_map(int16 dx, int16 dy){
/* Interpreta el contenido de un bloque de datos en RRAM y dibuja el mapa digital
de dicho bloque con un desplazamiento +/-dx y +/-dy.
*/
//:::::::::::::::::::::::::::::::::::::::::::::::::::://
```


ANEXO 5
MENSAJES DEL SISTEMA
(Ventanas de aviso)

Mensajes del sistema (ventanas de aviso)

No.	Mensaje	Significado	Tipo	Fuente
1	Card not ready or not present	Tarjeta no está lista ó no está presente	Alerta	SD
2	The card is unformatted	La tarjeta se encuentra sin formato	Alerta	SD
3	Format disk / All data will be lost, continue?	Formato de disco / Todos los datos serán borrados, continuar?	Alerta	SD
4	Format disk / Erasing CSI	Formato de disco / Borrando bloques CSI	Información	SD
5	Format disk / Erasing FDZ	Formato de disco / Borrando bloques de capa FDZ	Información	SD
6	Format disk / Done.	Formato de disco / Completado.	Información	SD
7	Delete file ?	Borrar Archivo?	Confirmación	SD
8	Not enough memory	No existe memoria suficiente	Alerta	SD
9	File not found	No se encuentra el archivo	Alerta	SD
10	Receive file / Retry?	Recepción de archivo / Reintentar (falla de conexión con PC)	Confirmación	COM
11	Receive file / Waiting...	Recepción de archivo / Esperando... (paquetes desde PC)	Información	COM
12	Receive file / Transmission Ended	Recepción de archivo / Transmisión finalizada (desde PC)	Información	COM
13	Receive file / Receive data /data packs: #	Recepción de archivo / Recibiendo paquetes Xmodem : #	Información	COM
14	Receive file / Total packs: #	Recepción de archivo / Total de paquetes Xmodem recibidos: #	Información	COM
15	GPS receiver / Initializing	Receptor GPS / Inicializando	Información	GPS
16	GPS receiver / Not ready!	Receptor GPS / No está listo (no existe señal en puerto serial)	Alerta	GPS
17	GPRMC frame lost or GPS not ready!	Receptor GPS / Trama GPRMC perdida o el receptor no está listo	Alerta	GPS
18	System failure	Falla general del sistema	Alerta	SYS
19	(-) LCD bright (+) / Level:	Control de brillo de pantalla / Nivel	Información	SYS

Los mensajes se clasifican en tres tipos: Alerta, información ó confirmación. Dependiendo de la fuente de accionamiento, estos pueden ser: SD (referido a la tarjeta de memoria SD ó también llamado disco), COM (referido a la comunicación con la PC), GPS (referido al receptor GPS), y SYS (referido al sistema en general).

ANEXO 6
ENTRADAS / SALIDAS DEL
MICROCONTROLADOR
(Pin-Out Quick Guide)

ENTRADAS / SALIDAS DEL MICROCONTROLADOR

GNOM-4550 Microcontroller Pin-out Quick Guide

MICROCONTROLLER SIDE						PERIPHERAL SIDE			
Type	Pin #	Port	Name	Function	Destination	Type	Pin #	Name	Function
i	2	RA0	AD_IN1	ADC Input 1	LM358	o	7	ANALOG	Analog External Input
i	3	RA1	AD_IN2	ADC Input 2				BAT_SIG	Battery Charge Signal
i	4	RA2	VREF-	ADC Negative Ref.					
i	5	RA3	VREF+	ADC Positive Ref.					
o	6	RA4	RS_Rx	RS-232 Receive	MAX232	i	9	RS_2UC	TTL Transceiver Input
i	7	RA5	RS_Tx	RS-232 Transmit	MAX232	o	10	UC_2RS	TTL Transceiver Output
i	33	RB0	SD_DI	SDI Data Input	SD Card Socket	o	7	DO	SD Card Data Out
o	34	RB1	SD_CK	SDI Clock	SD Card Socket	i	5	CLK	SD Card Clock
o	35	RB2	GD_RS	GLCD RS Select	GLCD	i	4	RS	GLCD RS Select
o	36	RB3	GD_RW	GLCS RW Select	GLCD	i	5	RW	GLCS RW Select
o	37	RB4	GD_E	GLCD Enable	GLCD	i	6	E	GLCD Enable
o	38	RB5	GD_CS1	GLCD CS1 Select	GLCD	i	15	CS1	GLCD CS1 Select
o	39	RB6	GD_CS2	GLCD CS2 Select	GLCD	i	16	CS2	GLCD CS2 Select
o	40	RB7	GD_RST	GLCD RST Signal	GLCD	i	17	RST	GLCD RST Signal
i	15	RC0	DN_ARROW	Down Arrow Input	Keyboard	o	1	DN_BTN	Down Arrow Keypad
i	16	RC1	LL_ARROW	Left Arrow Input	Keyboard	o	2	LL_BTN	Left Arrow Keypad
o	17	RC2	PWM_OUT	PWM Signal Output		i			
i	23	RC4	UP_ARROW	Up Arrow Input	Keyboard	o	3	UP_BTN	Up Arrow Keypad
i	24	RC5	RR_ARROW	Right Arrow Input	Keyboard	o	4	RR_BTN	Right Arrow Keypad
i	25	RC6	A_BUTTON	A Key Input	Keyboard	o	5	A_BTN	A Button Keypad
o	26	RC7	SD_DO	SDI Data Output	SD Card Socket	i			
i/o	19	RD0	GD_D0	GLCD Data D0	GLCD	i/o	7	DB0	GLCD Data D0
i/o	20	RD1	GD_D1	GLCD Data D1	GLCD	i/o	8	DB1	GLCD Data D1
i/o	21	RD2	GD_D2	GLCD Data D2	GLCD	i/o	9	DB2	GLCD Data D2
i/o	22	RD3	GD_D3	GLCD Data D3	GLCD	i/o	10	DB3	GLCD Data D3
i/o	27	RD4	GD_D4	GLCD Data D4	GLCD	i/o	11	DB4	GLCD Data D4
i/o	28	RD5	GD_D5	GLCD Data D5	GLCD	i/o	12	DB5	GLCD Data D5
i/o	29	RD6	GD_D6	GLCD Data D6	GLCD	i/o	13	DB6	GLCD Data D6
i/o	30	RD7	GD_D7	GLCD Data D7	GLCD	i/o	14	DB7	GLCD Data D7
o	8	RE0	SD_CS	SD Card Select	SD Card Socket	i		CS	SD Card Select
o	9	RE1	B_MUX	Mux Control B	4052	i		B_MUX	Mux Control B
o	10	RE2	A_MUX	Mux Control A	4052	i		A_MUX	Mux Control A
i	1	RE3	B_BUTTON	B Key Input	Keyboard	o	6	B_BTN	B Button Keypad

ANEXO 7
CARACTERÍSTICAS
SISTEMA DE ARCHIVOS FDZ

RESUMEN DE CARACTERÍSTICAS SISTEMA DE ARCHIVOS FDZ

Tamaño de disco			
	Bloques	Unidades	Bytes
Área de sistema	901	No Disponible	461,312
Área de usuario	1,764,808 1,734,896 (efectivos)	29912	903,581,696 888,266,752 (efectivos)
Total	1,765,709	No Disponible	904,043,008

Características del sistema	
Nombre del sistema	FDZ16
Tamaño de puntero	16 bits
Nombre del volumen	8 caracteres
Número de serie del volumen	15 dígitos
Número máximo de archivos	900
Tamaño máximo de archivo	7,2MB
Nombre de archivo	12 caracteres + extensión
Capacidad máxima de direccionamiento	65536 unidades / 1,966,080 bytes

Las características que se resumen, pertenecen al sistema de archivos FDZ desarrollado para un disco (memoria SD) de 1GB.

ANEXO 8
HOJAS TÉCNICAS
(*Datasheets*)

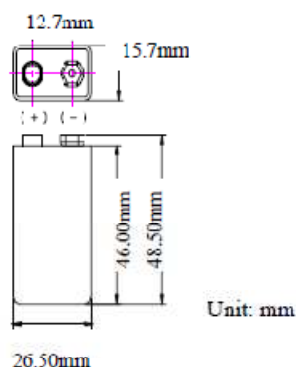
GP Batteries

Data Sheet

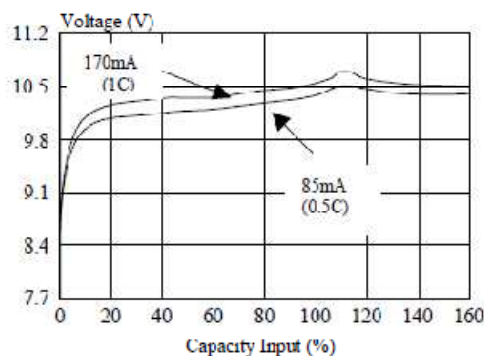
Model No.: GP17R8H

Type	Rechargeable Nickel Metal Hydride Cylindrical CdI
Nominal Dimension (with Sleeve)	15.7 mm(l) x 26.5 mm(w) x 48.5 mm(h)
Applications	Recommended discharge current 17 to 510 mA
Nominal Voltage	8.4V
Capacity	Minimum: 170 mAh Typical: 175 mAh when discharged at 0.2C to 1.0V at 20°C
Charging Condition	17 mA for 16 hrs at 20°C
Fast Charge	85 mA to 170 mA (0.5 to 1C) charge termination control recommended control parameters: -ΔV : 0 - 35mV DT/dt ** : 0.8°C/min (0.5 to 0.9C) : 0.8 - 1°C/min (1C) TCO ** : 45 - 50°C Timer ** : 105% nominal input ** For reference only
Service Life	>500 cycles (IEC standard) Up to 1000 cycle (not less than 60% of Nominal at 0.1C charge / 0.2C Discharge)
Continuous Overcharge	17 mA maximum current for 1 year No conspicuous deformation and/or leakage
Weight	32.0 g
Internal Resistance	Maximum 2000 mΩ upon fully charged at 1000I _Z

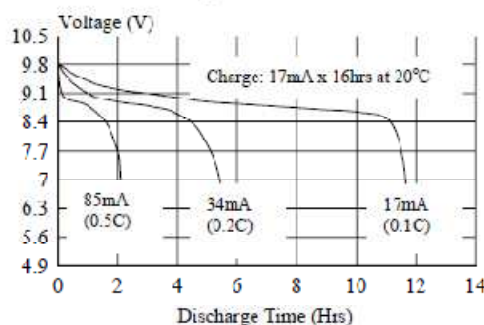
Max. Charging Voltage	10.5V at 17 mA charging
Ambient Temperature Range	Standard Charging : 0 to 45°C Fast Charging : 10 to 45°C Discharging : -20 to 50°C Storage : -20 to 35°C



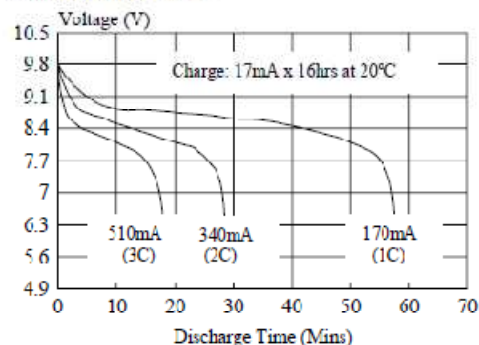
Fast Charge (charge control required)



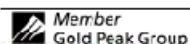
Low Rate Discharge



High Rate Discharge



* The information (subject to change without prior notice) contained in this document is for reference only and should not be used as a basis for product guarantee or warranty. For applications other than those described here, please consult your nearest GP Sales and Marketing Office or Distributors



Manufacturer reserves the right to alter or amend the design, model and specification without prior notice.

IRS0204 Rev.11


MICROCHIP
PIC18F2455/2550/4455/4550

28/40/44-Pin, High-Performance, Enhanced Flash, USB Microcontrollers with nanoWatt Technology

Universal Serial Bus Features:

- USB V2.0 Compliant
- Low Speed (1.5 Mb/s) and Full Speed (12 Mb/s)
- Supports Control, Interrupt, Isochronous and Bulk Transfers
- Supports up to 32 Endpoints (16 bidirectional)
- 1-Kbyte Dual Access RAM for USB
- On-Chip USB Transceiver with On-Chip Voltage Regulator
- Interface for Off-Chip USB Transceiver
- Streaming Parallel Port (SPP) for USB streaming transfers (40/44-pin devices only)

Power-Managed Modes:

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Idle mode currents down to 5.8 μ A typical
- Sleep mode currents down to 0.1 μ A typical
- Timer1 Oscillator: 1.1 μ A typical, 32 kHz, 2V
- Watchdog Timer: 2.1 μ A typical
- Two-Speed Oscillator Start-up

Flexible Oscillator Structure:

- Four Crystal modes, including High Precision PLL for USB
- Two External Clock modes, up to 48 MHz
- Internal Oscillator Block:
 - 8 user-selectable frequencies, from 31 kHz to 8 MHz
 - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @ 32 kHz
- Dual Oscillator options allow microcontroller and USB module to run at different clock speeds
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if any clock stops

Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
 - Capture is 16-bit, max. resolution 5.2 ns ($T_{CY}/16$)
 - Compare is 16-bit, max. resolution 83.3 ns (T_{CY})
 - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
 - Multiple output modes
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Enhanced USART module:
 - LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I²C™ Master and Slave modes
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

Special Microcontroller Features:

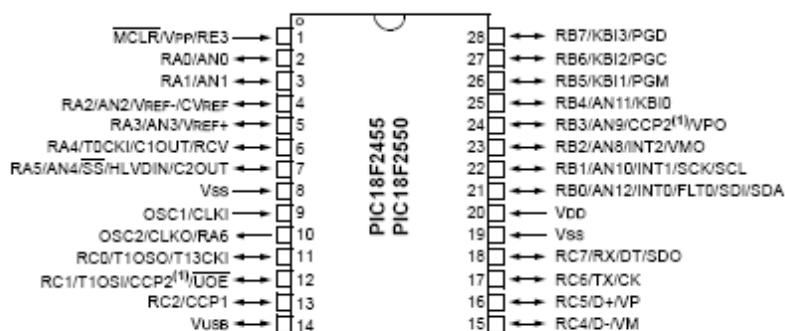
- C Compiler Optimized Architecture with optional Extended Instruction Set
- 100,000 Erase/Write Cycle Enhanced Flash Program Memory typical
- 1,000,000 Erase/Write Cycle Data EEPROM Memory typical
- Flash/Data EEPROM Retention: > 40 years
- Self-Programmable under Software Control
- Priority Levels for Interrupts
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
 - Programmable period from 41 ms to 131s
- Programmable Code Protection
- Single-Supply 5V In-Circuit Serial Programming™ (ICSP™) via two pins
- In-Circuit Debug (ICD) via two pins
- Optional dedicated ICD/ICSP port (44-pin devices only)
- Wide Operating Voltage Range (2.0V to 5.5V)

Device	Program Memory		Data Memory		I/O	10-Bit A/D (ch)	CCP/ECCP (PWM)	SPP	MSSP		EA/USART	Comparators	Timers 8/16-Bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)					SPI	Master I ² C™			
PIC18F2455	24K	12288	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F2550	32K	16384	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F4455	24K	12288	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3
PIC18F4550	32K	16384	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3

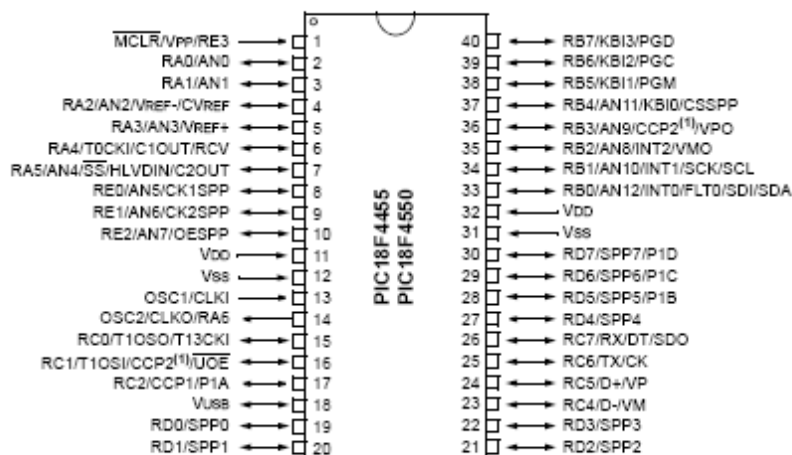
PIC18F2455/2550/4455/4550

Pin Diagrams

28-Pin PDIP, SOIC



40-Pin PDIP



Note 1: RB3 is the alternate pin for CCP2 multiplexing.

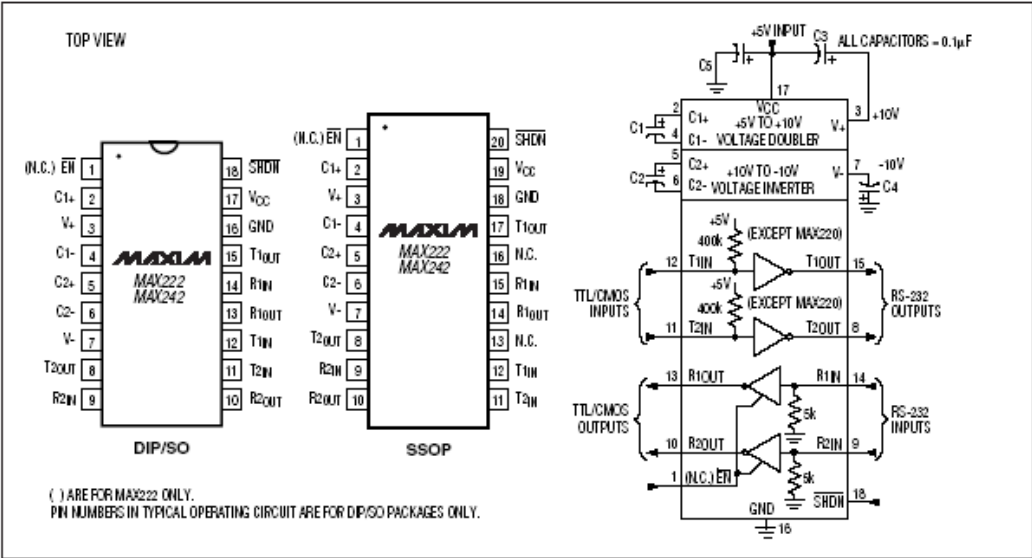
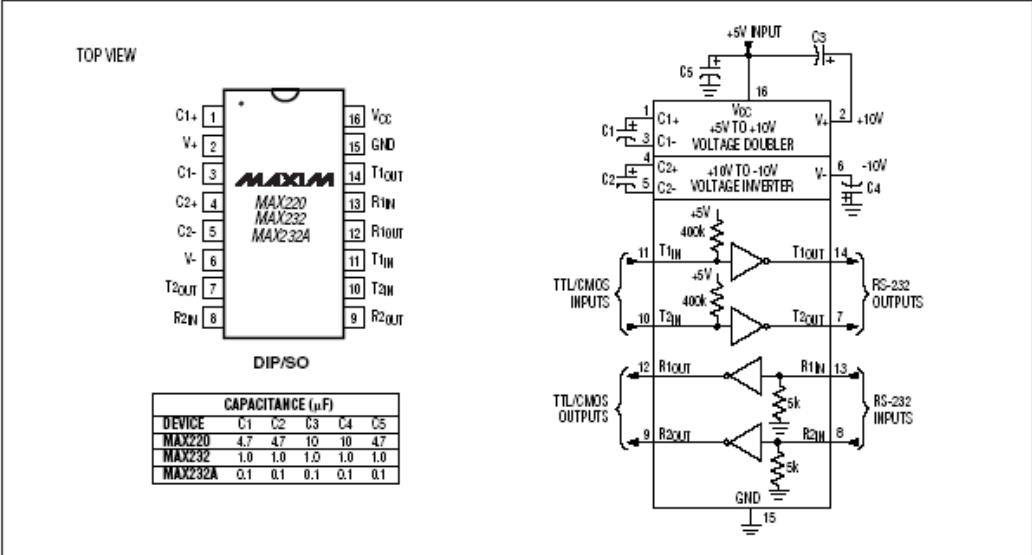
PIC18F2455/2550/4455/4550

TABLE 1-1: DEVICE FEATURES

Features	PIC18F2455	PIC18F2550	PIC18F4455	PIC18F4550
Operating Frequency	DC – 48 MHz	DC – 48 MHz	DC – 48 MHz	DC – 48 MHz
Program Memory (Bytes)	24576	32768	24576	32768
Program Memory (Instructions)	12288	16384	12288	16384
Data Memory (Bytes)	2048	2048	2048	2048
Data EEPROM Memory (Bytes)	256	256	256	256
Interrupt Sources	19	19	20	20
I/O Ports	Ports A, B, C, (E)	Ports A, B, C, (E)	Ports A, B, C, D, E	Ports A, B, C, D, E
Timers	4	4	4	4
Capture/Compare/PWM Modules	2	2	1	1
Enhanced Capture/ Compare/PWM Modules	0	0	1	1
Serial Communications	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART
Universal Serial Bus (USB) Module	1	1	1	1
Streaming Parallel Port (SPP)	No	No	Yes	Yes
10-Bit Analog-to-Digital Module	10 Input Channels	10 Input Channels	13 Input Channels	13 Input Channels
Comparators	2	2	2	2
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT
Programmable Low-Voltage Detect	Yes	Yes	Yes	Yes
Programmable Brown-out Reset	Yes	Yes	Yes	Yes
Instruction Set	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled
Packages	28-pin PDIP 28-pin SOIC	28-pin PDIP 28-pin SOIC	40-pin PDIP 44-pin QFN 44-pin TQFP	40-pin PDIP 44-pin QFN 44-pin TQFP

+5V-Powered, Multichannel RS-232 Drivers/Receivers

MAX220-MAX249



+5V-Powered, Multichannel RS-232 Drivers/Receivers

ABSOLUTE MAXIMUM RATINGS—MAX220/222/232A/233A/242/243

Supply Voltage (V _{CC})	-0.3V to +6V	20-Pin Plastic DIP (derate 8.00mW/°C above +70°C)	440mW
Input Voltages		16-Pin Narrow SO (derate 8.70mW/°C above +70°C)	696mW
T _{IN}	-0.3V to (V _{CC} - 0.3V)	16-Pin Wide SO (derate 9.52mW/°C above +70°C)	762mW
R _{IN} (Except MAX220)	±30V	18-Pin Wide SO (derate 9.52mW/°C above +70°C)	762mW
R _{IN} (MAX220)	±25V	20-Pin Wide SO (derate 10.00mW/°C above +70°C)	800mW
T _{OUT} (Except MAX220) (Note 1)	±15V	20-Pin SSOP (derate 8.00mW/°C above +70°C)	640mW
T _{OUT} (MAX220)	±13.2V	16-Pin CERDIP (derate 10.00mW/°C above +70°C)	800mW
Output Voltages		18-Pin CERDIP (derate 10.53mW/°C above +70°C)	842mW
T _{OUT}	±15V	Operating Temperature Ranges	
R _{OUT}	-0.3V to (V _{CC} + 0.3V)	MAX2_AC_, MAX2_C_	0°C to +70°C
Driver/Receiver Output Short Circuited to GND	Continuous	MAX2_AE_, MAX2_E_	-40°C to +85°C
Continuous Power Dissipation (T _A = +70°C)		MAX2_AM_, MAX2_M_	-55°C to +125°C
16-Pin Plastic DIP (derate 10.53mW/°C above +70°C)	842mW	Storage Temperature Range	-65°C to +160°C
18-Pin Plastic DIP (derate 11.11mW/°C above +70°C)	889mW	Lead Temperature (soldering, 10sec)	+300°C

Note 1: Input voltage measured with T_{OUT} in high-impedance state, SHDN or V_{CC} = 0V.

Note 2: For the MAX220, V₊ and V₋ can have a maximum magnitude of 7V, but their absolute difference cannot exceed 13V.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243

(V_{CC} = +5V ± 10%, C1-C4 = 0.1μF, MAX220, C1 = 0.047μF, C2-C4 = 0.33μF, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
RS-232 TRANSMITTERS						
Output Voltage Swing	All transmitter outputs loaded with 3kΩ to GND		±5	±8		V
Input Logic Threshold Low				1.4	0.8	V
Input Logic Threshold High	All devices except MAX220		2	1.4		V
	MAX220: V _{CC} = 5.0V		2.4			
Logic Pull-Up/Input Current	All except MAX220, normal operation			5	40	μA
	SHDN = 0V, MAX222/242, shutdown, MAX220			±0.01	±1	
Output Leakage Current	V _{CC} = 5.5V, SHDN = 0V, V _{OUT} = ±15V, MAX222/242			±0.01	±10	μA
	V _{CC} = SHDN = 0V, V _{OUT} = ±15V			±0.01	±10	
Data Rate				200	116	kb/s
Transmitter Output Resistance	V _{CC} = V ₊ = V ₋ = 0V, V _{OUT} = ±2V		300	10M		Ω
Output Short-Circuit Current	V _{OUT} = 0V		±7	±22		mA
RS-232 RECEIVERS						
RS-232 Input Voltage Operating Range					±30	V
RS-232 Input Threshold Low	V _{CC} = 5V	All except MAX243 R2 _{IN}	0.8	1.3		V
		MAX243 R2 _{IN} (Note 2)	-3			
RS-232 Input Threshold High	V _{CC} = 5V	All except MAX243 R2 _{IN}		1.8	2.4	V
		MAX243 R2 _{IN} (Note 2)		-0.5	-0.1	
RS-232 Input Hysteresis	All except MAX243, V _{CC} = 5V, no hysteresis in shdn.		0.2	0.5	1	V
	MAX243			1		
RS-232 Input Resistance			3	5	7	kΩ
TTL/CMOS Output Voltage Low	I _{OUT} = 3.2mA			0.2	0.4	V
TTL/CMOS Output Voltage High	I _{OUT} = -1.0mA		3.5	V _{CC} - 0.2		V
TTL/CMOS Output Short-Circuit Current	Sourcing V _{OUT} = GND		-2	-10		mA
	Sinking V _{OUT} = V _{CC}		10	30		



August 1999

LM117/LM317A/LM317 3-Terminal Adjustable Regulator

General Description

The LM117 series of adjustable 3-terminal positive voltage regulators is capable of supplying in excess of 1.5A over a 1.2V to 37V output range. They are exceptionally easy to use and require only two external resistors to set the output voltage. Further, both line and load regulation are better than standard fixed regulators. Also, the LM117 is packaged in standard transistor packages which are easily mounted and handled.

In addition to higher performance than fixed regulators, the LM117 series offers full overload protection available only in IC's. Included on the chip are current limit, thermal overload protection and safe area protection. All overload protection circuitry remains fully functional even if the adjustment terminal is disconnected.

Normally, no capacitors are needed unless the device is situated more than 6 inches from the input filter capacitors in which case an input bypass is needed. An optional output capacitor can be added to improve transient response. The adjustment terminal can be bypassed to achieve very high ripple rejection ratios which are difficult to achieve with standard 3-terminal regulators.

Besides replacing fixed regulators, the LM117 is useful in a wide variety of other applications. Since the regulator is "floating" and sees only the input-to-output differential volt-

age, supplies of several hundred volts can be regulated as long as the maximum input to output differential is not exceeded, i.e., avoid short-circuiting the output.

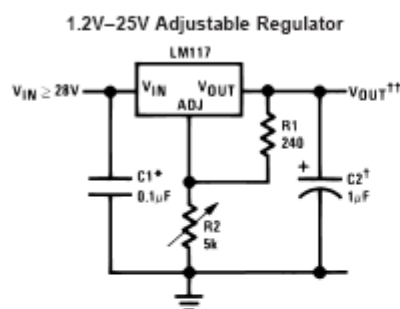
Also, it makes an especially simple adjustable switching regulator, a programmable output regulator, or by connecting a fixed resistor between the adjustment pin and output, the LM117 can be used as a precision current regulator. Supplies with electronic shutdown can be achieved by clamping the adjustment terminal to ground which programs the output to 1.2V where most loads draw little current.

For applications requiring greater output current, see LM150 series (3A) and LM138 series (5A) data sheets. For the negative complement, see LM137 series data sheet.

Features

- Guaranteed 1% output voltage tolerance (LM317A)
- Guaranteed max. 0.01%/V line regulation (LM317A)
- Guaranteed max. 0.3% load regulation (LM117)
- Guaranteed 1.5A output current
- Adjustable output down to 1.2V
- Current limit constant with temperature
- P* Product Enhancement tested
- 80 dB ripple rejection
- Output is short-circuit protected

Typical Applications



Full output current not available at high input-output voltages

*Needed if device is more than 6 inches from filter capacitors.

†Optional—improves transient response. Output capacitors in the range of 1 μ F to 1000 μ F of aluminum or tantalum electrolytic are commonly used to provide improved output impedance and rejection of transients.

$$V_{OUT} = 1.25V \left(1 + \frac{R_2}{R_1} \right) + I_{ADJ}(R_2)$$

LM117 Series Packages

Part Number Suffix	Package	Design Load Current
K	TO-3	1.5A
H	TO-39	0.5A
T	TO-220	1.5A
E	LCC	0.5A
S	TO-263	1.5A
EMP	SOT-223	1A
MDT	TO-252	0.5A

SOT-223 vs D-Pak (TO-252) Packages



Scale 1:1



June 2002

LM2936-3.3 Ultra-Low Quiescent Current 3.3V Regulator

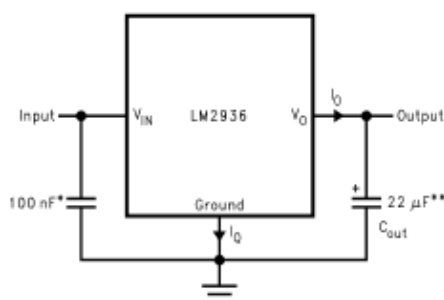
General Description

The LM2936-3.3 ultra-low quiescent current regulator features low dropout voltage and low current in the standby mode. With less than 20 μA quiescent current at a 100 μA load, the LM2936-3.3 is ideally suited for automotive and other battery operated systems. The LM2936-3.3 retains all of the features that are common to low dropout regulators including a low dropout PNP pass device, short circuit protection, reverse battery protection, and thermal shutdown. The LM2936-3.3 has a 40V maximum operating voltage limit, a -40°C to $+125^{\circ}\text{C}$ operating temperature range, and $\pm 3\%$ output voltage tolerance over the entire output current, input voltage, and temperature range. The LM2936-3.3 is available in a TO-92 package, a SO-8 surface mount package, as well as SOT-223 and TO-252 surface mount power packages.

Features

- Ultra low quiescent current ($I_Q \leq 20 \mu\text{A}$ for $I_O = 100 \mu\text{A}$)
- Fixed 3.3V, 50 mA output
- $\pm 2\%$ Initial output tolerance
- $\pm 3\%$ Output tolerance over line, load, and temperature
- Dropout voltage typically 200 mV @ $I_O = 50 \text{ mA}$
- Reverse battery protection
- -50V reverse transient protection
- Internal short circuit current limit
- Internal thermal shutdown protection
- 40V operating voltage limit
- 60V operating voltage limit for LM2936HV
- Shutdown pin available with LM2936BM package

Typical Application

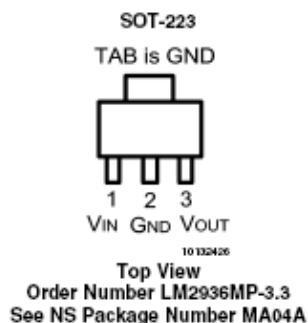
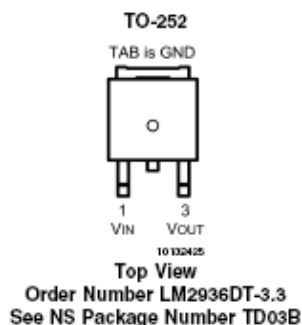


10132401

* Required if regulator is located more than 2" from power supply filter capacitor.

** Required for stability. Must be rated for 22 μF minimum over intended operating temperature range. Effective series resistance (ESR) is critical, see curve. Locate capacitor as close as possible to the regulator output and ground pins. Capacitance may be increased without bound.

Connection Diagrams





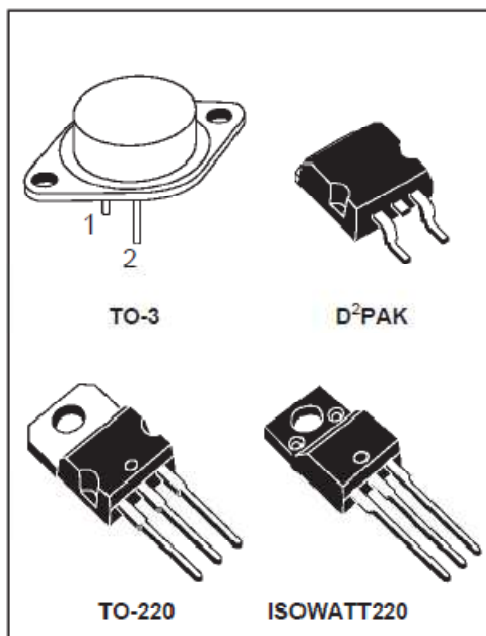
L7800 SERIES

POSITIVE VOLTAGE REGULATORS

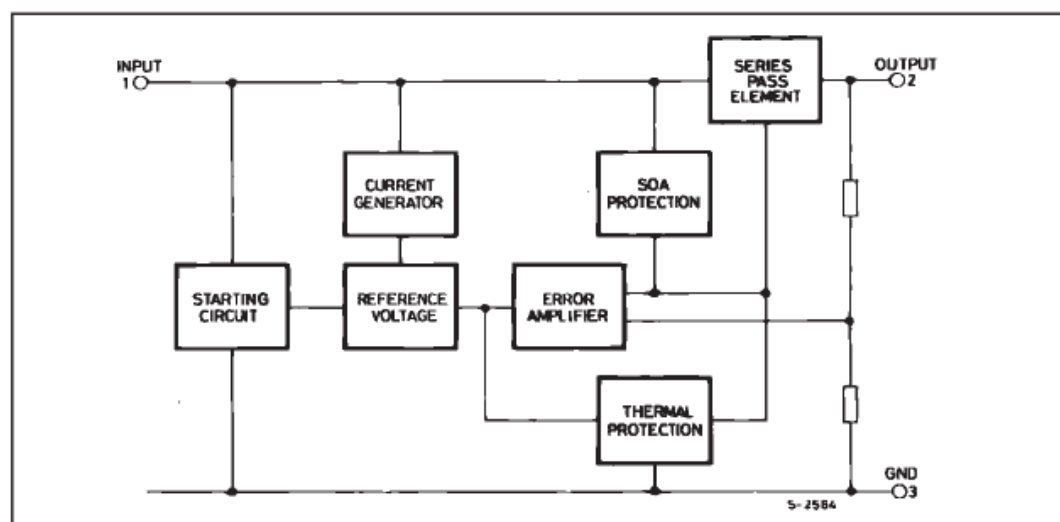
- OUTPUT CURRENT UP TO 1.5 A
- OUTPUT VOLTAGES OF 5; 5.2; 6; 8; 8.5; 9; 12; 15; 18; 24V
- THERMAL OVERLOAD PROTECTION
- SHORT CIRCUIT PROTECTION
- OUTPUT TRANSITION SOA PROTECTION

DESCRIPTION

The L7800 series of three-terminal positive regulators is available in TO-220 ISOWATT220 TO-3 and D²PAK packages and several fixed output voltages, making it useful in a wide range of applications. These regulators can provide local on-card regulation, eliminating the distribution problems associated with single point regulation. Each type employs internal current limiting, thermal shut-down and safe area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.

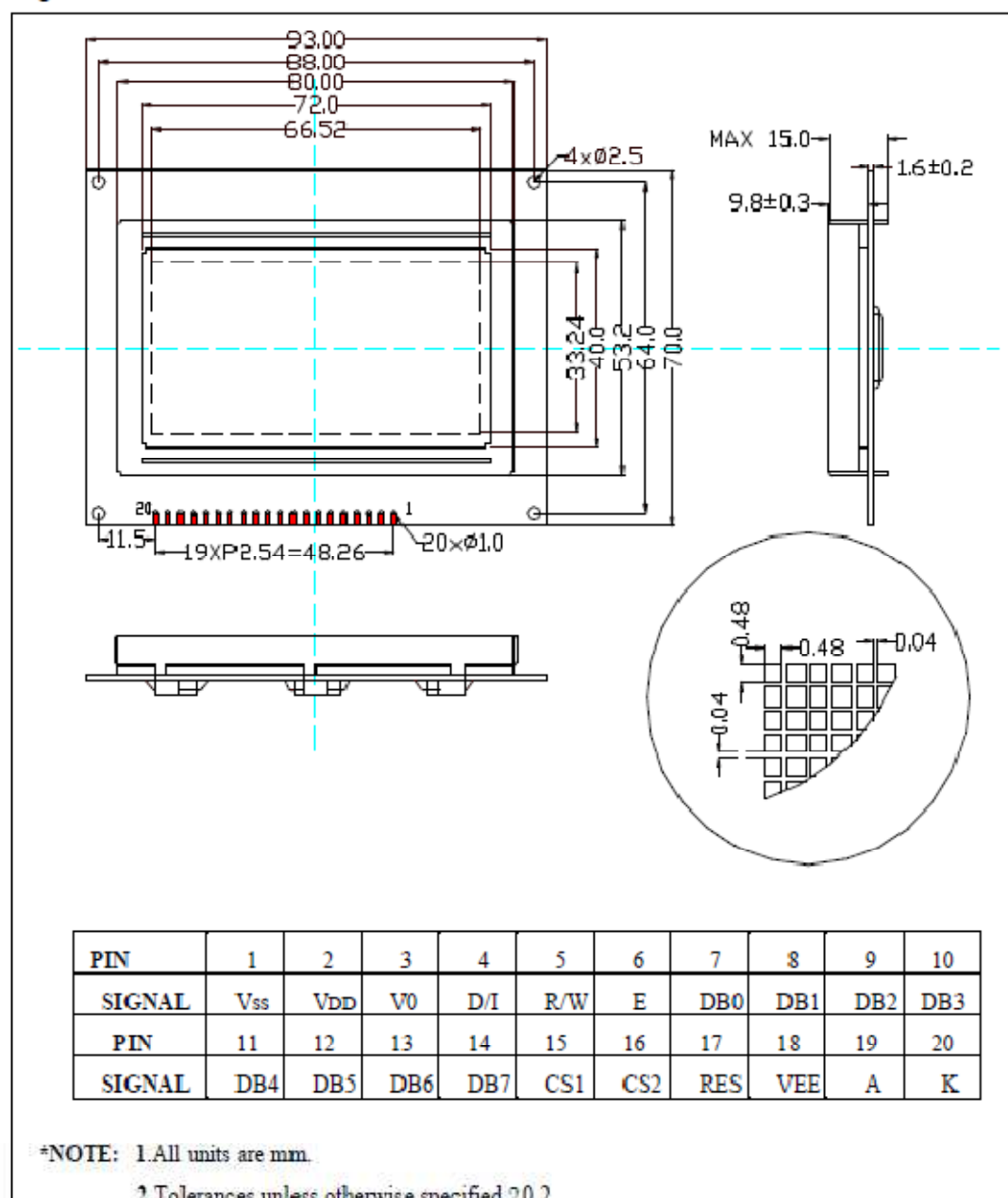


BLOCK DIAGRAM



Características físicas y especificación de pines del TS12864B

Figure 1. External Dimensions



RS232 GPS receiver

Product Feature

- "SiRF Star II e/LP" high performance and low power consumption chipset
- All-in-view 12-channel parallel processing
- Built-in active antenna
- High sensitivity to satellite signal
- Cold start under 45 seconds, average
- Superior urban canyon performance
- FoliageLock for weak signal tracking
- Build-in SuperCap to reserve system data for rapid satellite acquisition.
- Supported NMEA 0183 command: GGA, GSA, GSV, RMC, GLL, VTG
- Super-cohesive magnetic for mounting on the car
- LED indicator for GPS fix or not fix
 - LEDOFF: Receiver switch off
 - LED ON: No fixed, Signal searching
 - LED Flashing: Position Fixed
- Non-slip on the bottom
- RS232 interface with PS2 connector

System Specification

Electrical Characteristics (Receiver)	
Frequency	L1, 1575.42 MHz
C/A Code	1.023 MHz chip rate
Channels	12
Sensitivity	-170 dBW
Accuracy	
Position Horizontal	10m 2D RMS (SA off)
Velocity	0.1m/sec 95% (SA off),
Time	1 micro-second synchronized to GPS time
WAAS enabled	7m 2D RMS
Datum	
Datum	WGS-84
Acquisition Rate	
Hot start	8 sec., average (with ephemeris and almanac valid)
Warm start	38 sec., average (with almanac but not ephemeris)
Cold start	45 sec., average (neither almanac nor ephemeris)
Reacquisition	0.1 sec. average (interruption recovery time)
Protocol	
GPS Protocol	Default: NMEA 0183 (Secondary: SiRF binary)
GPS Output Data	SiRF binary >> position, velocity, altitude, status and control ; NMEA 0183 protocol, and supports command: GGA, GSA, GSV, RMC, VTG, GLL (VTG and GLL are optional)
GPS transfer rate	Software command setting (Default : 4800,n,8,1 for NMEA)
Dynamic Condition	
Acceleration Limit	Less than 4g

	Altitude Limit	18,000 meters (60,000 feet) max.
	Velocity Limit	515 meters/sec. (1,000 knots) max.
	Jerk Limit	20 m/sec**3
Temperature		
	Operating	-20°~ 80°C
	Storage	-30°~ 85°C
	Humidity	Up to 95% non-condensing
Power		
	Voltage	5V ±5%
	Current	80mA typical (Continuous mode)
Physical Characteristics		
	Dimension	2.32" x 1.85" x 0.82" (59mm x 47mm x 21mm)
	Cable Length	60" (1500mm)
Low Noise Amp		
	Amplifier Gain w/out cable	27 dB Typical
	Filtering	-25dB (+100 MHz)
	Output VSWR	2.0 Max.
	Voltage	DC 3 ~ 5.0V
	Current	15mA max. @5VDC



RS232 GPS Receiver

Due to continuous product improvements, all specifications may be subject to change without notice

REFERENCIAS BIBLIOGRÁFICAS

LIBROS

1. Bao-Yen Tsui, James, *Fundamentals of Global Positioning Systems Receivers*, Segunda Edición, Editorial John Wiley & Sons Inc, 2006.
2. Hofmann-Wellenhof, Legat, Wieser, Lichtenegger, *Navigation: Principles of Positioning and Guidance*, Primera Edición, Editorial Springer, 2003.

INTERNET

1. Mapas raster y vector,
www.usna.edu/Users/oceano/pguth/website/so432web/rastervector.htm, Enero 2010.
2. Mapas vectoriales INOCAR,
www.inocar.mil.ec, Enero 2010.
3. Estructura FAT,
<http://home.teleport.com/~brainy/fat32.htm>,
<http://www.persystems.net/sosvirus/pregunta/mbr.htm>, Febrero 2010.
4. Historia y evolución GPS,
<http://homepages.mty.itesm.mx/al584299/mypaper.htm>,
<http://www.alsitel.com/tecnico/gps/historia.htm>,
<http://www.tomtom.com/howdoesitwork/page.php?ID=9&CID=2&Language=1>,
<http://www.clubdelamar.org/sistemagps.htm>,
<http://www.codepedia.com/1/The+GPRMC+Sentence>,

- <http://personal.redestb.es/jatienza/gps/datos.htm>,
<http://www.spacearchive.info/utc.htm>,
5. Funcionamiento, definiciones, trama de datos GPS,
<http://personal.redestb.es/jatienza/gps/datos.htm>,
http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=1271,
 6. Especificación bus SPI,
http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus, Febrero 2010.
 7. Códigos ASCII,
<http://www.portalplanetasedna.com.ar/ascii.htm>, <http://www.asciitable.com/>,
 8. Pantallas gráficas LCD,
<http://www.dinceraydin.com/lcd/gfxintro.htm>, Marzo 2010.
 9. Especificaciones de tarjetas SD,
www.sdcard.org, Marzo 2010.
 10. Protocolo RS232,
<http://perso.wanadoo.es/pictob/comserie.htm>,
 11. Compilador CCS,
<http://www.ccsinfo.com/content.php?page=compilers>, Abril 2010.
 12. Microcontrolador PIC18F4550,
<http://ww1.microchip.com/downloads/en/DeviceDoc/39632b.pdf>, Febrero 2010.
 13. Eagle 3D y Povray,
<http://felixchenier.com/doku.php?id=pcb:eagle3dnewpart>,
<http://www.matwei.de/doku.php?id=en:eagle3d:eagle3d>, Mayo 2010.
 14. Protocolo NMEA0183,
www.tronico.fi/OH6NT/docs/NMEA0183.pdf,
http://www.marimsys.com/paginas/nmea_codigo.htm, Mayo 2010.

15. Recursos Visual Basic,

<http://www.canalvisualbasic.net/manual/inicio-visual-basic/estructuras-datos/>,
<http://msdn.microsoft.com/es-es/library>,
[http://msdn.microsoft.com/es-es/library/wz100x8w\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/wz100x8w(VS.80).aspx),
<http://www.recursosvisualbasic.com.ar/htm/tutoriales/tutorial-basico7.htm>,
<http://www.recursosvisualbasic.com.ar/htm/trucos-codigofuente-visual-basic/36.htm>,
<http://www.recursosvisualbasic.com.ar/htm/tutoriales/tutorial-basico6.htm>,
<http://www.ms-news.net/f1181/la-funcion-para-truncar-en-vbulletin-10180092.html>,
http://www.edudevices.com.ar/download/articulos/visualbasic/CURSO_VB_07.pdf,
<http://www.mcanam.com/articulos/VB.php?id=1>,
<http://www.canalvisualbasic.net/manual/manual-visual-basic/acceso-datos/>,
<http://www.tecnun.es/asignaturas/Informat1/AyudaInf/aprendainf/VisualBasic6/basic60.pdf>, Agosto del 2010.

16. Archivos KML,

http://earth.google.es/userguide/v4/ug_kml.html
<http://www.tufuncion.com/kmz-google-earth>
http://earth.google.com/intl/es/userguide/v4/ug_kml.html, Agosto del 2010.

17. Definiciones técnicas,

http://www.unicrom.com/Tut_rms_promedio.asp
<http://www.alegsa.com.ar/Dic/fat.php>
<http://www.webopedia.com/TERM/D/DDR4M.html>,

HOJAS TÉCNICAS Y ARTÍCULOS TÉCNICOS

1. United Nations Secretariat - Department of Economic and Social Affairs, Integration of GPS Digital Imagery and GIS with Census Mapping , September 2004, Febrero 2010.
2. Globalsat, Product Specification - Rs232 Gps Receiver, Febrero 2010.
3. STmicroelectronics, L7800 Series Positive Voltage Regulators, 1998, Marzo 2010.

4. GP Batteries, Data Sheet Model No. GP17R8H - TRS0204 Rev.11, Marzo 2010.
5. Intech LCD Group Ltd, User's Guide GDM12864A (Liquid Crystal Display Module), 1998, Febrero 2010.
6. Hantronix, Interfacing a Hantronix 128x64 Graphic Module to an 8-bit Microcontroller Application Note, Febrero 2010.
7. Samsung Electronics, KS0108B 64ch Segment Driver For Dot Matrix LCD, Febrero 2010.
8. National Semiconductor, LM117/LM317A/LM317 3-Terminal Adjustable Regulator, 1999, Febrero 2010.
9. Maxim, +5V-Powered, Multichannel RS-232 Drivers/Receivers, Agosto 2001, Febrero 2010.
10. Microchip, PIC18F2455/2550/4455/4550 Data Sheet, Febrero 2010.
11. Techstar Electronics Co. Ltd, TS12864B Spec Datasheet, Febrero 2010.
12. Compsys, Storage Card Interface Kit for MultiMediaCards (MMC) and Secure Digital Cards (SD), 2007, Marzo 2010.
13. Sandisk Corporation, Host Design Considerations: NAND MMC and SD-Based Products Application Note Version 1.0, September 2002, Marzo 2010.
14. Hitachi, Multimediacard 16Mbyte/32Mbyte/64Mbyte/128Mbyte, April 2002, Marzo 2010.
15. Sandisk Corporation, Sandisk Secure Digital Card Product Manual Version 1.9 December 2003, Abril 2010.
16. SD Group Matsushita Electric Industrial Co. Ltd, SD Memory Card Specifications -Simplified Version Of Physical Layer Specification Version 1.01, April 2001, Abril 2010.
17. SD Group Matsushita Electric Industrial Co. Ltd, SD Memory Card Specifications - File System Specification Version 1.0, February 2000, Abril 2010.
18. SD Association, SD Card Specification - Simplified Version of: Part E1 Secure Digital Input/Output (SDIO) Card Specification Version 1.00, October 2001, Abril 2010.
19. Technical Committee SD Association, SD Specifications Part A2 SD Host Controller Simplified Specification Version 2.00, February 8, 2007, Abril 2010.
20. Toshiba Corporation, TOSHIBA SD Card Specification, Febrero 2003, Abril 2010.

21. Narváez V. Carlos A., Protocolos de Transferencia de Archivos - Xmodem Transferencia de Archivos PC – PIC, 2003, Mayo 2010.
22. National Semiconductor, LM2936-3.3 Ultra-Low Quiescent Current 3.3V Regulator, June 2002, Mayo 2010.

LEGALIZACIÓN

Sangolquí, Enero del 2011

Documento elaborado por:

Paúl Enrique Nicolalde Ávila

Ing. Hugo Ortiz T. Msc.
DIRECTOR DEL DEPARTAMENTO

Ab. Jorge Carvajal
SECRETARIO ACADÉMICO

ESCUELA POLITÉCNICA DEL EJÉRCITO

CARRERA DE INGENIERÍA ELECTRÓNICA EN

TELECOMUNICACIONES

Fecha de entrega:.....

Paúl Enrique Nicolalde Ávila

Ing. Gonzalo Olmedo, Ph.D.
COORDINADOR DE CARRERA