

ESCUELA POLITÉCNICA DEL EJÉRCITO

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA,
AUTOMATIZACIÓN Y CONTROL

PROYECTO DE GRADO PARA LA OBTENCIÓN DEL TÍTULO EN
INGENIERÍA

SISTEMA DE REGISTRO DE EVENTOS EN RUTA PARA
TRANSPORTACIÓN PÚBLICA

Autores

Marcelo Fernando Paredes Solano

Luís Alfredo Poaquiza Yumbolema

Sangolquí – Ecuador

2007

CERTIFICACIÓN

Certificamos que el siguiente proyecto de grado titulado “SISTEMA DE REGISTRO DE EVENTOS EN RUTA PARA TRANSPORTACIÓN PÚBLICA” ha sido desarrollado en su totalidad por el Sr. Marcelo Fernando Paredes Solano con C.I. 171704644-3, y el Sr. Luís Alfredo Poaquiza Yumbolema con C.I. 050228613-1 bajo nuestra dirección.

Ing. Paúl Ayala
DIRECTOR

Ing. Alejandro Chacón
CODIRECTOR

AGRADECIMIENTO

A todas las personas que de una u otra forma me han ayudado en el transcurso de mi vida y forman parte de mi vida.

A mis padres, por ser mi apoyo y guía durante toda mi vida, que han sido mi soporte, por darme la fuerza y el valor necesarios para siempre salir siempre adelante, por enseñarme a enfrentar los problemas de forma directa, “Gracias” por todo papis por ustedes he logrado esto.

A mi hermana que me ha brindado su ayuda cuando lo he necesitado.

A mis primos Nelly y Fabricio por haber sido mi apoyo en muchas ocasiones, por haberme guiado, por ser como son.

A mi familia por haberme apoyado y dado los consejos oportunos para salir adelante.

A todos mis profesores en las distintas etapas de mi vida estudiantil, por todos los conocimientos que me supieron inculcar.

A mi amigo y compañero Luís que me brindo su apoyo y comprensión para concluir nuestro objetivo.

A mi Director y Codirector, por su ayuda en la consecución de esta meta.

Marcelo Fernando Paredes Solano

AGRADECIMIENTO

Son muchas las personas a quienes debo agradecer por el apoyo

Quiero agradecer primero a Dios, que con su amor y misericordia me acompañó día a día para conseguir este anhelado objetivo, brindándome fuerza en los momentos más difíciles de mi vida.

A mis padres, quienes desde niño me inculcaron el valor de la superación, de la amistad, humildad y sobre todo del respeto. Y a quienes les doy mis mas sinceros agradecimientos y un Dios les Pague por ayudarme a culminar con este titulo.

A mis hermanas y hermanos quienes con alegría, supieron llenar mi corazón de esperanza, y orgullo de tener una familia en quien apoyarme.

A mi enamorada Vicky, quien es mi fuerza, mis ganas de luchar y sobre todo una de las participes para conseguir mis metas.

A mis amigos y amigas de la U, quienes me brindaron su amistad incondicional, gracias a todos ellos por ser parte de mi familia.

A mis amigos y amigas de la Pastoral Juvenil de San Felipe, especialmente a las Hermanas Betlehemitas, Hermanas Franciscanas de la Juventud, y Catequistas que con su carisma y cariño me supieron dar a conocer a Jesús y María, y a seguir sus pasos.

A mi compañero y amigo Marcelo, que junto a su familia, supieron ser personas generosas y luchadoras.

A mi Director y Codirector, quienes fueron también protagonistas de la consecución de este proyecto.

A todos los profesores de la ESPE y amigos que de una u otra manera estuvieron presentes en un instante de mi vida.

Muchas Gracias....

Luís Poaquiza

DEDICATORIA

A *mi madre* por su apoyo incondicional, por sus consejos, por su coraje, por su motivación que me ha permitido seguir adelante, por la cual me encuentro en este mundo, por ser mi amiga..

A *mi padre* por su ejemplo de constancia y honradez que lo caracterizan y me ha infundado siempre, por ser mi amigo, por haberme guiado.

A *mi hermana* por estar a mi lado en los buenos y malos momentos tratando de ayudarme a resolver mis dudas y problemas.

A *mi abuelito* por ser una persona luchadora, por sus consejos, por ser mi amigo, por su ayuda incondicional.

A *mis amigos* por haberme brindado su ayuda y consejos.

Este proyecto es para todos ustedes.

Marcelo Fernando Paredes Solano

DEDICATORIA

Este proyecto esta dedicado con mucho aprecio a las personas que forman parte de mi vida.

A mi Dios, que a pesar de todas mis equivocaciones, él me ha perdonado, y ha continuado dando salud y vida.

A mis padres, quienes han hecho posible realizar este proyecto, y son ellos el apoyo y soporte de la familia.

A mis hermanos y hermanas, quienes se encuentran en la etapa de estudiantes, y aliento a que superen todas las barreras que la vida nos pone en nuestro camino.

A mi enamorada quien es la persona que me acompaño en este duro camino y a la vez vida Universitaria.

A mis amigos, de quienes he aprendido a valorar a las personas y son ejemplo de superación.

A mis Profesores de la ESPE, que tengan presente que los conocimientos impartidos nos ha servido para este proyecto y sobre todo para ser líderes y protagonistas de una nueva sociedad.

Con cariño:

Luís Poaquiza

PRÓLOGO

El desarrollo e implementación del “Sistema de Registro de Eventos en Ruta para transportación Pública” tiene como propósito el de proveer a las autoridades respectivas de una herramienta de registro y control, con el fin de contar con una herramienta que ayude a evitar posibles accidentes de tránsito.

El “Sistema de Registro de Eventos en Ruta para Transportación Pública” se encuentra concebido en el sensamiento de variables como velocidad de vehículo conforme un rango de seguridad y de la posición obtenida por medio de un GPS, con éstos dos registros el procesador, una vez concluido el trayecto de viaje, almacena los datos en un CD-ROM a ser entregado a las autoridades respectivas para su análisis. Los datos almacenados de velocidad y posición son procesados por un software que permitirá desplegar las curvas de comportamiento de las variables almacenadas, de ésta manera se obtendrán datos fiables que permitirán determinar excesos de velocidad en ruta, su posición e incluso el tiempo al que ocurrió. De ocurrir un accidente los datos se encontrarán almacenados en una memoria interna del sistema que pueden ser extraídos con fines investigativos.

TABLA DE CONTENIDOS

1	INTRODUCCIÓN	1
2	DESCRIPCION OPERATIVA Y FUNCIONAL DEL SISTEMA DE REGISTRO DE EVENTOS EN RUTA	6
2.1	SISTEMA DE ALIMENTACION.....	7
2.2	SISTEMA DE ADQUISICION DE DATOS.....	7
2.3	SISTEMA DE ALMACENAMIENTO DE DATOS.....	9
2.4	SISTEMA DE GRABADO CD–RW.....	9
2.5	SISTEMA DE INTERPRETACION DE DATOS.....	10
2.6	UNIDADES DE ALMACENAMIENTO ÓPTICO, SD CARD, MICROCONTROLADORES, SENSORES Y GPS.....	10
2.7	UNIDAD DE ALMACENAMIENTO ÓPTICO CD-RW.....	11
2.7.1	<i>Principio de Funcionamiento de CD-RW</i>	11
2.7.2	<i>Grabación de CD</i>	12
2.7.2.1	Encabezamiento del CD.....	12
2.7.2.2	Formato XA.....	12
2.7.2.3	Extended attribute records (XAR).....	14
2.7.2.4	Técnicas de Grabación en CD.....	14
2.7.3	<i>Buses de Datos</i>	15
2.7.3.1	Organización de la comunicación del bus.....	16
2.8	SD CARD.....	17
2.8.1	<i>Características de la SD CARD</i>	18
2.8.1.1	Características generales.....	18
2.8.1.2	Características físicas de la SD CARD.....	18
2.8.1.3	Características eléctricas de la SD CARD.....	19
2.8.2	<i>Modos de Operación</i>	20
2.8.2.1	Modo SD.....	20
2.8.2.2	Modo SPI.....	21
2.8.3	<i>Organización de la Tarjeta</i>	22
2.9	MICROCONTROLADOR.....	24
2.9.1	<i>Microcontrolador PIC16F877A</i>	24
2.9.1.1	Programación.....	25
2.10	SENSORES.....	25
2.10.1	<i>Interruptor Reed (Magnético)</i>	26
2.10.2	<i>Montaje</i>	26
2.10.3	<i>Protección</i>	27
2.10.3.1	NEMA.....	27
2.10.3.2	DIN.....	28
2.11	GPS (SISTEMA DE POSICIONAMIENTO GLOBAL).....	29
2.11.1	<i>Principio de Funcionamiento</i>	29
2.11.1.1	Mensaje de Navegación.....	30
2.11.2	<i>Componentes del Sistema</i>	33
2.11.2.1	Sector Espacial.....	33
2.11.2.2	Sector de Control.....	34
2.11.2.3	Sector de Usuarios.....	34
2.11.3	<i>Equipo de Observación</i>	35
2.11.3.1	La antena de recepción.....	35
2.11.3.2	El sensor.....	35

2.11.3.3	El controlador	37
2.11.4	<i>Errores y Aplicaciones futuras</i>	38
2.11.4.1	Errores en GPS	38
2.11.4.2	Aplicaciones Futuras	39
3	PROTOCOLOS DE COMUNICACIÓN Y SOFTWARE	41
3.1	INTRODUCCION	41
3.2	PROTOCOLOS DE COMUNICACIÓN	44
3.2.1	<i>Serial Peripheral Interface (SPI)</i>	44
3.2.1.1	Estructura del protocolo SPI.....	45
3.2.1.2	Modo de Transmisión y recepción de información.	47
3.2.1.3	Configuración e inicialización en modo SPI de la SD Card y del PIC 16F877A.	48
3.2.1.4	Lectura y escritura de datos en la memoria SD Card.....	51
3.2.2	<i>Universal Synchronous Asynchronous Receiver Transmitter (USART)</i>	52
3.2.2.1	Protocolo de comunicación serie.....	52
3.2.2.2	Configuración de la USART del PIC16F877	55
3.2.3	<i>Attachment with Packet Interface (ATAPI/IDE)</i>	58
3.2.3.1	ATAPI Protocolo	58
3.2.3.2	Registros del un dispositivo ATAPI.....	59
3.2.3.3	Atapi Packet Command.....	62
3.2.3.4	Transferencia de comandos al dispositivo ATAPI	63
3.2.3.5	Escritura en un CD	64
3.2.3.6	Lectura de un CD	66
3.2.4	<i>IDE interface</i>	67
3.2.5	<i>Protocolo NMEA</i>	68
3.2.5.1	Estructura de la trama NMEA para sentencia RMC (Recommend Minimum Specific Data) 70	70
3.2.5.2	Habilitación del puerto serial del GPS	70
3.2.5.3	Selección y control de velocidad de transmisión de trama	71
3.2.5.4	Selección de Datos.....	72
3.3	SOFTWARE	73
3.3.1	<i>GPL GNU (General Public Licence)</i>	73
3.3.2	<i>GAMBAS</i>	74
3.3.2.1	Desarrollo	76
3.3.2.2	Elementos de Gambas	76
3.3.2.3	Portabilidad.....	77
3.3.2.4	Multiplataforma	77
3.3.2.5	Cómo obtener Gambas	77
3.3.2.6	Diferencias con Visual Basic (6.0)	77
4	DISEÑO E IMPLEMENTACIÓN DEL SISTEMA DE CONTROL DE EVENTOS EN RUTA..	79
4.1	INTRODUCCIÓN	79
4.2	DESCRIPCIÓN DEL SISTEMA DEL SISTEMA IMPLEMENTADO	80
4.3	DISEÑO DE HARDWARE.....	81
4.3.1	<i>Generación de la señal de reloj (clock)</i>	81
4.3.2	<i>Circuito de visualización de información</i>	82
4.3.3	<i>Circuito de conexión del GPS</i>	83
4.3.4	<i>Circuito de conexión de la tarjeta SD Card</i>	84
4.3.5	<i>Circuito de conexión para el dispositivo ATAPI/IDE</i>	85
4.3.6	<i>Circuito de comunicación Serial</i>	87
4.3.7	<i>Circuito de conexión del sensor de velocidad</i>	87
4.3.8	<i>Circuito de la fuente de Alimentación</i>	88
4.3.9	<i>Interfaz de Usuario</i>	89
4.4	PROGRAMACIÓN DEL MICROCONTROLADOR (PIC16F877A)	91
4.4.1	<i>Inicialización de puertos comunicaciones y puertos físicos</i>	91
4.4.2	<i>Programa de visualización del LCD</i>	92
4.4.2.1	Función de Inicialización del LCD.....	92
4.4.2.2	Funciones de escritura al LCD	93
4.4.2.3	Funciones goto y clear del LCD.....	94
4.4.3	<i>Función de retardo (Delay)</i>	95
4.4.4	<i>Función de Inicialización del USART del PIC16F877A</i>	95
4.4.5	<i>Funciones de transmisión y recepción serial por el puerto USART</i>	96
4.4.6	<i>Inicialización de la SD Card o MMC</i>	96

4.4.7	<i>Transmisión de Datos a la SD Card</i>	97
4.4.8	<i>Envío de comandos a la SD Card</i>	97
4.4.9	<i>Lectura de un byte de la SD Card y transmitir</i>	98
4.4.10	<i>Escritura en una dirección de la memoria SD card</i>	98
4.4.11	<i>Inicialización y detección de un dispositivo ATAPI</i>	100
4.4.12	<i>Envío del packet command (A0) al dispositivo ATAPI</i>	101
4.4.13	<i>Escritura de datos en un CD</i>	101
4.4.14	<i>Configuración e inicialización del Módulo GPS</i>	102
4.4.15	<i>Configuración de USART en el PIC16F877A y Separación de trama GPS</i>	104
4.5	DESARROLLO DE SOFTWARE	108
4.5.1	DISEÑO HMI	108
4.5.1.1	<i>Pantalla de Presentación</i>	109
4.5.1.2	<i>Pantalla de Acceso</i>	110
4.5.1.3	<i>Pantalla de Principal</i>	111
5	ANÁLISIS E INSTALACIÓN DEL PROTOTIPO	118
5.1	INTRODUCCIÓN	118
5.2	DATOS TÉCNICOS DEL PROTOTIPO	118
5.2.1	<i>Alimentación</i>	118
5.2.2	<i>Consumo de corriente</i>	118
5.2.3	<i>Tiempo de muestreo de datos</i>	119
5.2.4	<i>Transferencia de Datos a CD-RW</i>	119
5.3	INSTALACIÓN DEL SISTEMA PROTOTIPO EN EL VEHICULO DE PRUEBA	119
5.3.1	INSTALACIÓN DEL PROTOTIPO	119
5.3.1.1	<i>Alimentación</i>	119
5.3.1.2	<i>Controlador y CD-RW</i>	121
5.3.1.3	<i>Interfaz de usuario</i>	121
5.3.1.4	<i>Sensor de Velocidad</i>	122
5.3.1.5	<i>Antena de GPS</i>	122
5.3.2	INICIO DEL SISTEMA Y ADQUISICIÓN DE DATOS	123
5.3.3	GRABACIÓN EN CD	124
5.3.4	TRANSFERENCIA DE DATOS A LA PC (Personal Computer)	124
6	CONCLUSIONES Y RECOMENDACIONES	128
7	BIBLIOGRAFÍA	132
8	ANEXOS	135
8.1	ANEXO 1. DIAGRAMA DE FLUJO DEL ENVÍO DEL PACKET COMMAND AL DISPOSITIVO	136
8.2	ANEXO 2. DIAGRAMA DE FLUJO DE COMANDOS QUE REQUIEREN LEER DATOS DEL DISPOSITIVO 142	
8.3	ANEXO 3. DIAGRAMA DE FLUJO DE COMANDOS QUE REQUIEREN ESCRIBIR DATOS AL DISPOSITIVO	145
8.4	ANEXO 4. DIAGRAMA DE FLUJO DE COMANDOS QUE NO REQUIEREN DATOS	148
8.5	ANEXO 5. LISTA DE ATA COMMANDS	149
8.6	ANEXO 6. CDROM ATAPI PACKET COMMAND	150
8.7	ANEXO 7. RECONOCIMIENTO DEL ESTADO DE UN CD	151
8.8	ANEXO 8. PROCESO DE ESCRITURA	152
8.9	ANEXO 9. MANUAL DE USUARIO	153

INDICE DE TABLAS

Tabla 1.1 Serie Histórica. Accidentes de Transito y victimas	1
Tabla 2.1 Rangos de voltaje y corriente SD CARD	19
Tabla 2.2 Aplicación de los Pines en el Modo SD	21
Tabla 2.3 Aplicación de los Pines en el Modo SPI.....	21
Tabla 2.4 Tipos de Formatos	23
Tabla 2.5 Organización lógica interna de la tarjeta.....	24
Tabla 2.6 Características del Microcontrolador PIC16F877A.....	25
Tabla 2.7 Especificaciones DIN	28
Tabla 2.8 Trama del Sistema de Transmisión GPS.....	32
Tabla 2.9 Mensajes de Salida del Estándar NEMA.....	32
Tabla 2.10 Detalles de la trama RMC.....	33
Tabla 3.1 Transferencia de datos del protocolo SPI.....	45
Tabla 3.2 Principales comandos del protocolo de acceso mediante SPI	46
Tabla 3.3 Respuestas R1 y RB	46
Tabla 3.4 Respuesta R2.....	47
Tabla 3.5 Generación de Baudios.....	56
Tabla 3.6 Registros de un dispositivo ATAPI.....	60
Tabla 3.7 Registro Status.....	60
Tabla 3.8 Registro error	61
Tabla 3.9 Parámetros de configuración para el command packet.....	63
Tabla 3.10 Pines del conector IDE.....	68
Tabla 3.11 Mensajes y tramas de salida del GPS	69
Tabla 3.12 Transporte de Mensaje	69
Tabla 3.13 Estructura del mensaje RMC	70
Tabla 3.14 Configuración del puerto serial GPS.....	71
Tabla 3.15 Control de velocidad y mensajes del GPS.....	72
Tabla 3.16 Mensajes de salida del GPS	72
Tabla 3.17 Selección de datos del GPS	73
Tabla 3.18 Características de Gambas.....	75
Tabla 4.1. Asignación de pines del PIC16F877A (2)	81
Tabla 4.2. Asignación de pines del PIC16F877A (1)	81
Tabla 4.3. Asignación de pines del LCD	82
Tabla 4.4. Asignación de pines del módulo GPS.....	83
Tabla 4.5. Asignación de pines de la memoria SD Card	84
Tabla 4.6. Asignación de pines del conector IDE	86
Tabla 5.1 Potencia consumida por el prototipo para la fuente de 5V.....	134
Tabla 5.2 Potencia consumida por el prototipo para la fuente de 12V.....	134
Tabla 5.3 Costo directo del prototipo.....	140

INDICE DE FIGURAS

Figura 1.1 Caja negra.....	3
Figura 1.2 Caja negra de un avión siniestrado.	4
Figura 1.3 Esquema General del Sistema de Registro de Eventos en Ruta para Transportación Pública.....	5
Figura 2.1 Esquema de señales sensadas, adquisición de datos (Velocidad y Posición).	6
Figura 2.2 Diagrama de Bloques del sistema de alimentación.....	7
Figura 2.3 Diagrama de Bloques del Sistema de Adquisición de Datos.....	7
Figura 2.4 Diagrama de Bloques (Adquisición de Datos “VELOCIDAD”).....	8
Figura 2.5 Diagrama de Bloques (Adquisición de Datos “POSICION”).....	8
Figura 2.6 Diagrama de Bloques (Sistema de Almacenamiento de Datos)	9
Figura 2.7 Diagrama de Bloques (Sistema de Grabado CD-RW)	9
Figura 2.8 Diagrama de Bloques (Sistema de Interpretación de Datos)	10
Figura 2.9 Esquema General del Sistema de Registro de Eventos.....	10
Figura 2.10 Esquema del bus de datos	15
Figura 2.11 Utilización del Bus de Datos	17
Figura 2.12 Distintos tipos de tarjetas de memoria Flash disponibles en el mercado. De izquierda a derecha: MemoryStick (Sony), XD (Olympus Optical y Fuji Photo Film), SD/MultiMediaCard , SmartMedia (Olympus Optical y Fujy Foto Film) , Compact Flash y MiniSD.....	18
Figura 2.13 Características físicas de la tarjeta	19
Figura 2.14 Características eléctricas de la tarjeta	19
Figura 2.15 Organización física interna de la tarjeta.....	22
Figura 2.16 Organización del mapa de memoria de la tarjeta	23
Figura 2.17 Sensor Inductivo	25
Figura 2.18 Sensor tipo REED.....	26
Figura 2.19 Sensores inductivos detección del número de vueltas y sentido de giro.	27
Figura 2.20 Posicionamiento de los Satélites GPS.....	30
Figura 2.21 Mensajes de Navegación.....	31
Figura 2.22 Componentes de un Sistema GPS	35
Figura 3.1 Comunicación de datos entre dispositivos.....	41
Figura 3.2 Transferencia de datos del protocolo SPI.....	42
Figura 3.3 Estructura de comandos SPI	42
Figura 3.4 Lectura de un bloque de la tarjeta	47
Figura 3.5 Escritura de un bloque a la tarjeta.	48
Figura 3.6 Transmisión serial	52
Figura 3.7 Codificación no auto reloj.....	54
Figura 3.8 Estructura de la trama serial del PIC16F877A	55
Figura 3.9 Estándar ATAPI/ATA	59
Figura 3.10 Registro de control.....	61
Figura 3.11 Registro Head device.....	62
Figura 3.12 Formato de comandos ATAPI.....	62
Figura 3.13 Proceso de escritura de un paquete.....	65
Figura 3.14 Formato del comando de escritura	65
Figura 3.15 Formato del comando de lectura	66
Figura 3.16 Conector IDE.....	67

Figura 4.1. Circuito de generación de señal de reloj.....	82
Figura 4.2. Circuito de visualización.....	82
Figura 4.3. Circuito de conexión del módulo GPS	83
Figura 4.4. Circuito de conexión de la memoria SD Card.....	85
Figura 4.5. Circuito de conexión para el dispositivo ATAPI/IDE	86
Figura 4.6. Circuito de comunicación serial.	87
Figura 4.7 Circuito de la señal del sensor de velocidad	88
Figura 4.8 Circuito de la fuente de alimentación.....	88
Figura 4.9 Circuito de Interfaz de Usuario	89
Figura 4.10 Visualización de Datos.....	90
Figura 4.11 Interfaz de Usuario.....	91
Figura 4.12. Flujograma de la inicialización del LCD.....	93
Figura 4.13. Flujograma del proceso de escritura de comando/caracter	94
Figura 4.14. Flujograma del proceso inicialización de la SD Card	97
Figura 4.15. Lectura de la memoria SD Card en modo SPI	98
Figura 4.16. Escritura en la memoria SD Card en modo SPI	99
Figura 4.17. Escritura en la memoria SD Card en modo SPI	100
Figura 4.18. Configuración e Inicialización del módulo GPS	102
Figura 4.19. Configuración del Puerto serial GPS (Set Main Serial Port).....	103
Figura 4.20. Cambio del protocolo SIRF a protocolo NMEA	103
Figura 4.21. Selección de la trama a transmitir y velocidad de la misma.....	103
Figura 4.22. Selección del tipo de Dato a transmitir (WSG 84)	104
Figura 4.23. Proceso general del GPS	106
Figura 4.24. Separación trama de GPS	107
Figura 4.25. Esquema de Software SSV 1.0	108
Figura 4.26. Pantalla de Presentación de SSV 1.0.....	109
Figura 4.27. Pantalla de Información SSV 1.0.....	109
Figura 4.28. Pantalla de Acceso SSV 1.0.....	110
Figura 4.29. Pantalla Principal SSV 1.0.....	111
Figura 4.30. Pantalla de Ingreso de Datos de Recorrido.....	112
Figura 4.31. Pantalla de Selección de Recorrido.....	113
Figura 4.32. Mensaje de Advertencia (Falta de Datos)	113
Figura 4.33. Pantalla de Selección de Recorridos	114
Figura 4.34. Mensaje de Advertencia (Falta de Placa).....	114
Figura 4.35. Opciones de Acceso a datos	115
Figura 4.36. Pantalla de Informes Gráficos.....	116
Figura 4.37. Presentación de Velocidades mayores a 70Km/h	117
Figura 5.1 Vehículo utilizado para instalación del prototipo	133
Figura 5.2 Dispositivos del Prototipo	135
Figura 5.3 Conexiones del prototipo.....	135
Figura 5.4 Conexión de Interfaz de usuario con conector	136
Figura 5.5 Conexión de Interfaz de usuario directa.....	136
Figura 5.6 Ubicación de la antena del GPS.....	137
Figura 5.7 Pantallas de Interfaz de Usuario.....	137
Figura 5.8 Pantalla de usuario para Grabar CD.....	138
Figura 5.9 Pantalla de transferencia de Datos.....	138
Figura 5.10 Esquema de variables y almacenamiento en el gestor de Base de Datos MySQL.....	139

CAPÍTULO I

INTRODUCCIÓN

Estos últimos años el Ecuador ha sido testigo de la infinidad de accidentes de tránsito que se han suscitado, posee uno de los índices más altos de accidentes según lo demuestran las estadísticas, debido a la falta de recursos de las autoridades para controlar el cumplimiento de la Ley de Tránsito.

AÑOS	ACCIDENTES	VÍCTIMAS
1.996	20.910	8.619
1.997	17.666	8.461
1.998	17.712	8.113
1.999	17.111	8.250
2.000	15.987	7.710
*2001	11.008	4.986
*2002	12.215	6.357
*2003	10.369	6.841
2.004	16.323	10.128
2.005	16.578	10.697
2.006	16.623	12.432
* NO INCLUYE LA PROVINCIA DEL GUAYAS		

Tabla 1.1 Serie Histórica. Accidentes de Tránsito y víctimas ocurridos en el Ecuador (1996-2006).

Fuente INEC

Los intentos por disminuir la cantidad de infracciones por exceso de velocidad por parte de las autoridades de tránsito son insuficientes; en el sistema actual, el control se ha venido realizando en forma externa al usuario mediante el uso de radares y patrullajes en ruta, lo que conlleva una inversión alta e insuficiente del Fisco por este concepto.

Durante la mayor parte de los últimos años los esfuerzos en investigación y desarrollo por parte de los fabricantes ha estado encaminado a que los autos sean mucho mas eficientes en cuanto a consumo de combustible tratando al mismo tiempo de hacerlos más seguros, no obstante el implementación de sistemas que permitan obtener registros de datos en una ruta determinada no han sido desarrollados.

Una vez vista la necesidad de obtener elementos de control de accidentes tanto persuasivos como preventivos, se ha planteado el desarrollo de un prototipo que permita obtener un registro de los eventos que se desarrollasen en una determinada ruta de viaje.

El sistema planteado se encuentra concebido en el sensamiento de variables como velocidad de vehículo conforme un rango de seguridad y de la posición obtenida por medio de un receptor GPS, con éstos dos registros el procesador, una vez concluido el trayecto de viaje, almacena los datos en un medio óptico a ser entregado a las autoridades respectivas para su análisis.

Los datos almacenados de velocidad y posición son procesados por un software que permitirá desplegar las curvas de comportamiento de las variables almacenadas, de ésta manera se obtendrán datos fiables que permitirán determinar excesos de velocidad en ruta, su posición e incluso el tiempo al que ocurrió el evento.

De ocurrir un accidente los datos se encontrarán almacenados en una memoria interna del sistema que pueden ser recuperados y analizados con fines investigativos.

La idea de crear este tipo de dispositivos viene precedida por las ya conocidas cajas negras de los aviones, que comenzaron a utilizarse en los primeros vuelos a finales de los años 50.



Figura 1.1 Caja negra.

Se trata de dos cajas: la CVR y la FDR. La primera se encarga de registrar todo el sonido que se produce en la cabina del avión y la segunda caja recoge información sobre los parámetros del trayecto.

En la actualidad estas cajas han evolucionado, desde sus inicios y **son capaces de registrar más de 350 parámetros diferentes** entre los que destacan la altitud y velocidad del vuelo, la aceleración vertical y destino, fallos electrónicos, las conversaciones de los pilotos o comunicaciones efectuadas por radio.

Los registradores actuales emplean microcircuitos de memoria flash, capaces de almacenar datos durante varios años sin alimentación de energía. Los mejores registradores de estado sólido pueden guardar del orden de 80 megabytes, mucho menos que la memoria de la mayoría de los ordenadores personales, pero suficiente para almacenar unas dos horas de grabaciones de voces de cabina o un día completo de lecturas de los instrumentos del avión.

Los registradores contienen también tarjetas de circuito que procesan y comprimen los datos, aunque sólo los microcircuitos de memoria están encerrados en el bloque antichoque de la caja.

Ese bloque se cubre con un blindaje grueso de acero para que resista los aplastamientos por impacto. Bajo el acero hay una capa de aislante térmico diseñado para proteger los microcircuitos de memoria de los incendios de alta temperatura que suelen iniciarse tras un accidente del reactor.



Figura 1.2 Caja negra de un avión siniestrado.

Las cajas están dentro de recipientes hechos de materiales muy resistentes, como el titanio y pueden soportar varias veces la fuerza de la gravedad y temperaturas superiores a los mil grados centígrados.

Las Cajas Negras soportan las siguientes Prueba de certificación:

- *Prueba de impacto:* una pistola de gas lanza el registrador contra un blanco de aluminio, produciendo una fuerza máxima de 3500 g.
- *Prueba de resistencia a la penetración:* se deja caer sobre el aparato, desde tres metros de altura, una masa de 225 Kg. provisto de una punta de acero templado.
- *Prueba de aplastamiento estático:* un actuador aplica una compresión de 2300 kilogramos fuerza.
- *Prueba de inmersión a gran profundidad:* el registrador debe resistir 24 horas en una cámara llena de agua marina a presión.
- *Prueba de ignifugación:* se le somete a llamas de 1100 °C.

Tomando en cuenta todas las especificaciones que se tiene para la fabricación de cajas de seguridad aéreas (“cajas negras”), se ha visto la posibilidad de desarrollar un sistema de monitoreo y registro de eventos que permita obtener las características que se dieron en un determinado viaje.

Por tanto se ha planteado la solución en el diseño de la siguiente manera:

El sistema a implementar debe almacenar la información para la ubicación, hora, fecha y el estado de velocidad, en un punto dado del recorrido de un vehículo de transporte pública o de carga.

Debe poseer la opción de grabar en un medio de almacenamiento óptico (CD), y este se pueda reproducir en un computador personal (PC).

La información total será almacenada en una base de datos y visualizada en un software amigable para el usuario.

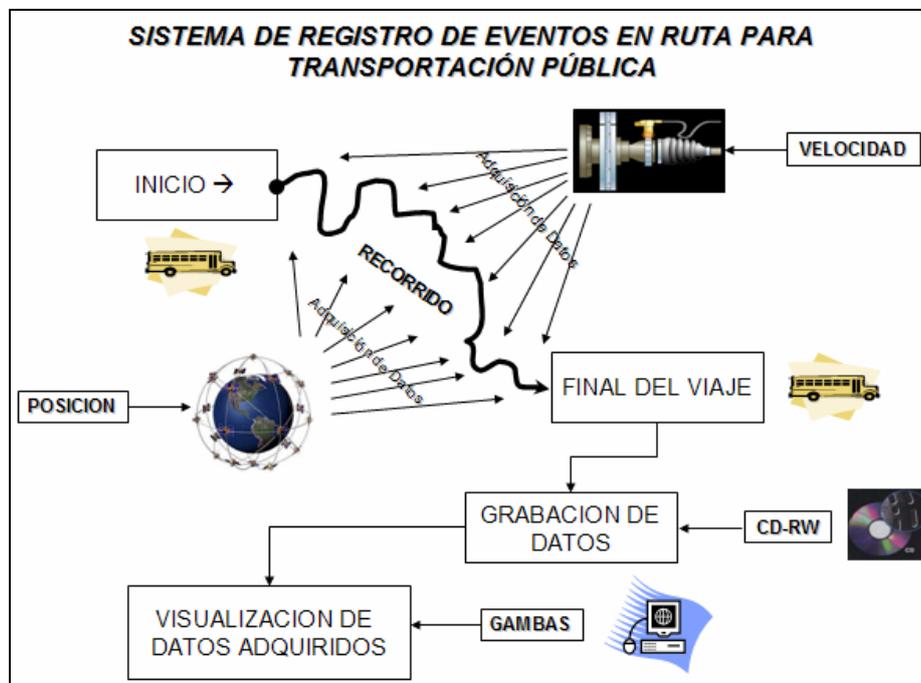


Figura 1.3 Esquema General del Sistema de Registro de Eventos en Ruta para Transportación Pública.

CAPITULO II

DESCRIPCIÓN OPERATIVA Y FUNCIONAL DEL SISTEMA DE REGISTRO DE EVENTOS EN RUTA.

Se ha considerado para el diseño del prototipo el sensamiento de dos señales; *velocidad y posición*, teniendo en cuenta que el sistema será diseñado de tal manera que pueda expandirse y adaptar nuevas variables al mismo para ser sensadas y registradas.

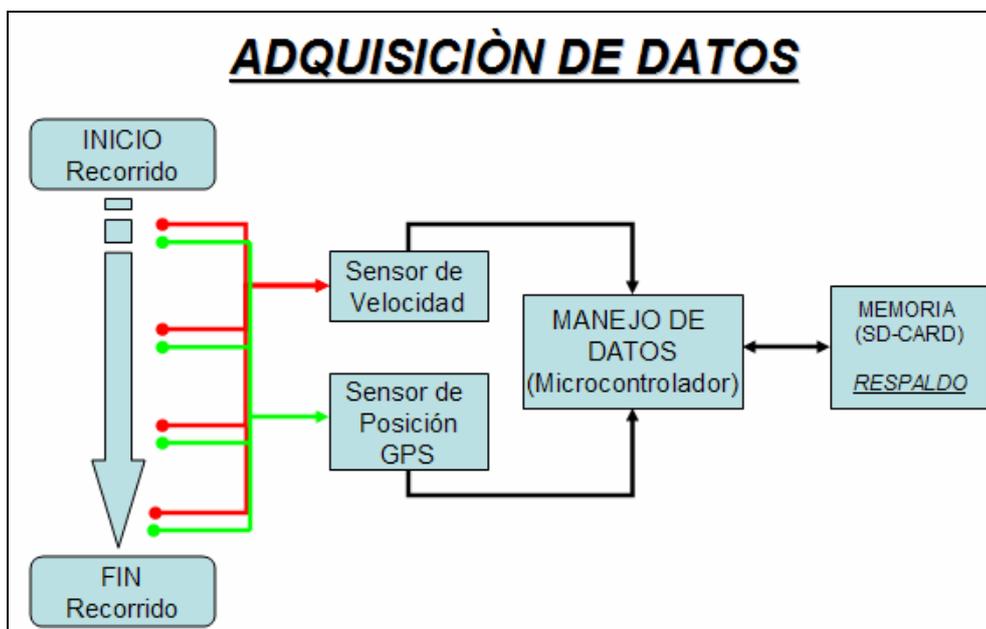


Figura 2.1 Esquema de señales sensadas, adquisición de datos (Velocidad y Posición).

El sistema de Registro de Eventos en Ruta para la Transportación Pública, se ha planteado en las siguientes etapas:

- Etapa 1.- Sistema de Alimentación.
- Etapa 2.- Sistema de Adquisición de Datos.
- Etapa 3.- Sistema de Almacenamiento de datos (Respaldo).
- Etapa 4.- Sistema de Grabado de Cd-Rw.
- Etapa 5.- Sistema de Interpretación de Datos (Software GAMBAS).

2.1 SISTEMA DE ALIMENTACION

El sistema de alimentación para el proyecto desarrollado, ha tomado en cuenta la salida de la Batería que todo automotor tiene, proveyéndonos de 12 voltios de corriente continua. Dado que el sistema de monitoreo necesita de alimentación en 5 voltios y en 12 voltios, el diseño se lo ha hecho tomando en cuenta estos dos valores.

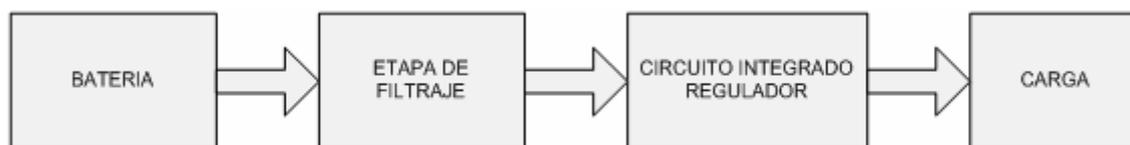


Figura 2.2 Diagrama de Bloques del sistema de alimentación.

2.2 SISTEMA DE ADQUISICION DE DATOS.

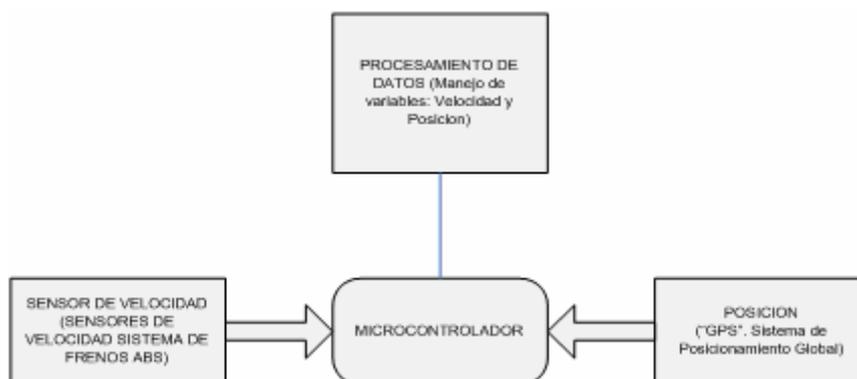


Figura 2.3 Diagrama de Bloques del Sistema de Adquisición de Datos.

La adquisición de datos para el Sistema de Registro de Eventos para Transportación Pública como primera etapa de desarrollo se ha tomado en cuenta el sensamiento de señales como *velocidad, posición y tiempo*, teniendo en cuenta que el sistema de registro tendrá la posibilidad de monitorear y registrar variables a través de una conexión al bus de datos de los automotores, permitiéndonos con esto evitar tener sensores redundantes.

La Adquisición de los parámetros de velocidad para el desarrollo del sistema han sido tomados de los Sensores de Velocidad provistos en los sistemas de Frenos, tomando en cuenta que se puede tener como un sistema redundante el parámetro de velocidad que el sistema de posicionamiento global puede proveer.

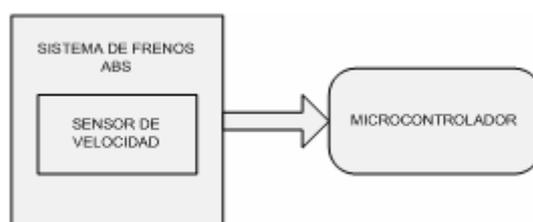


Figura 2.4 Diagrama de Bloques (Adquisición de Datos "VELOCIDAD")

La adquisición del dato "*Posición*" en el Sistema de Registro de Eventos para Transportación Pública se ha realizado utilizando un Receptor "GPS" que permite tener datos tanto de Longitud, Latitud y Orientación, datos con los cuales se puede obtener la ubicación en cualquier parte del Planeta.

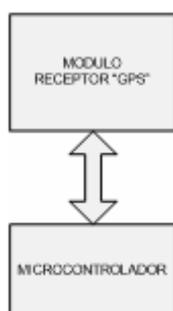


Figura 2.5 Diagrama de Bloques (Adquisición de Datos "POSICION")

2.3 SISTEMA DE ALMACENAMIENTO DE DATOS

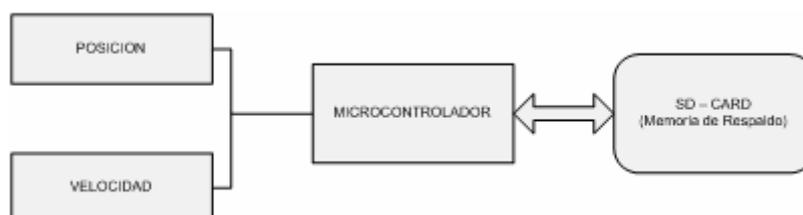


Figura 2.6 Diagrama de Bloques (Sistema de Almacenamiento de Datos)

El almacenamiento de Datos en el Sistema de Registro ha sido provisto de una memoria auxiliar que para el caso es una memoria SD-CARD, en la cual se esta guardando la información requerida (Velocidad, Posición, Tiempo), es importante considerar que el uso de la memoria SD-CARD también servirá como una memoria de respaldo a los datos obtenidos por el sistema, en caso de que sucediese un accidente y no se pudiera grabar los datos en el CD-RW.

La utilidad que se ha visto al utilizar la memoria SD-CARD es debido a que si se necesitase aumentar la capacidad de almacenamiento de datos del sistema y por ende alcanzar mayores distancias de recorrido con el respectivo registro de datos solo se debería aumentar la capacidad de almacenamiento de la memoria. El manejo de la memoria SD-CARD (lectura – escritura) se lo ha realizado utilizando el microcontrolador PIC16F877A.

2.4 SISTEMA DE GRABADO CD-RW.



Figura 2.7 Diagrama de Bloques (Sistema de Grabado CD-RW)

La implementación de un sistema que permitiese obtener un respaldo óptico del recorrido realizado, planteo el manejo de la unidad de CD-RW que permitiría guardar los datos obtenidos en un CD. La información recopilada durante el recorrido que fue almacenada por la memoria de respaldo SD-CARD, se grabará

en el CD por medio del CD-RW permitiendo obtener el respectivo respaldo óptico de la ruta realizada.

2.5 SISTEMA DE INTERPRETACION DE DATOS

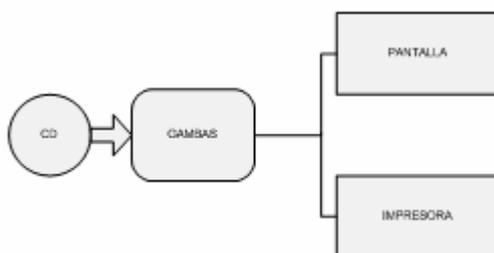


Figura 2.8 Diagrama de Bloques (Sistema de Interpretación de Datos)

El sistema brinda la posibilidad de interpretar los datos obtenidos por medio de graficas comparativas que permitirán analizar el recorrido realizado permitiendo interpretar las características del trayecto en su conducción.

2.6 UNIDADES DE ALMACENAMIENTO ÓPTICO, SD CARD, MICROCONTROLADORES, SENSORES Y GPS

La utilización de las unidades de almacenamiento óptico, sd card, microcontroladores, sensores y gps, relacionadas entre si, brinda la posibilidad de obtener las características requeridas para el diseño del sistema de registro. A continuación se muestra el diagrama de la relación existente entre los elementos anteriormente nombrados:

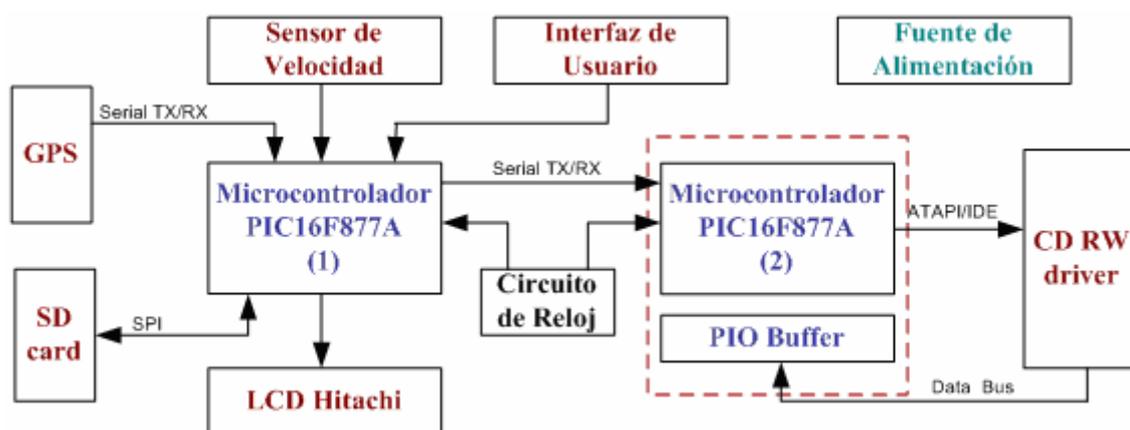


Figura 2.9 Esquema General del Sistema de Registro de Eventos.

2.7 UNIDAD DE ALMACENAMIENTO ÓPTICO CD-RW.

2.7.1 Principio de Funcionamiento de CD-RW.

CD-RW son las siglas de CD ReWritable, es decir CD re-escribible, asociado a la tecnología de regrabación por cambio de fase. También se denominan CD-E (CD-Erasable) o su equivalente CD borrable.

Esta tecnología se basa en la propiedad que posee una capa de material como el Telurio (mezclado con germanio o antimonio), de cambiar del estado amorfo (0) al cristalino (1) si se alcanza la "temperatura transición" (100 °C ó más); y de volver de cristalino a amorfo, si se alcanza la "temperatura de fusión" y se deja enfriar.

Para escribir un uno en un punto de una pista del disco, un láser con baja potencia lo calienta rápidamente hasta la temperatura de transición. Si el estado físico del punto era amorfo, pasa a cristalino; y si ya está en este estado, quedará igual. Un cero se escribe calentando el punto hasta la temperatura de fusión, usando el láser con alta potencia. Al enfriarse pasa al estado amorfo, y si estaba en ese estado volverá al mismo. La lectura de las pistas así grabadas se realiza con el mismo cabezal, recorriéndoles con el láser de Potencia diez veces menor. La luz láser reflejada al ser sensada permite detectar, por diferencias de reflectividad, los cambios de un estado físico al otro, a lo largo de la pista. Un punto en estado cristalino refleja el 70% de la luz incidente, y en estado amorfo el 18%.

Obsérvese que esta tecnología es puramente óptica, sin magnetismo, requiriéndose una sola pasada para escribir. Para escribir o leer este tipo de discos se requiere grabadoras y lectoras apropiadas para su tecnología. Se estimaba hace poco un CD-E puede regrabarse unas 100.000 veces. Realizando 50 reescrituras diarias, duraría 5 años (de 365 días). Ha habido avances al respecto. Las unidades CD-RW pueden también leer los CD-ROM y CD-R, siendo además que estos CD cumplen con el formato UDF (Universal Disc Format)

normalizado por la Asociación OSTA, que facilita a los sistemas operativos el acceso a discos.

2.7.2 Grabación de CD

2.7.2.1 Encabezamiento del CD

La superficie de grabación de un CD se divide en tres partes: el LEAD IN, la ZONA DE DATOS y el LEAD OUT.

El LEAD IN (el “encabezamiento”) ocupa los primeros cuatro milímetros del CD en el margen interior y contiene una especie de índice. A continuación sigue la zona de datos que ocupa 33mm.

La parte final la constituye la zona del LEAD OUT, que es una especie de marca final. Se encuentra inmediatamente detrás del final de la zona de datos ocupada y tiene una anchura de 1mm.

2.7.2.2 Formato XA

La principal característica del estándar XA (eXtended Architecture) es su capacidad de interdireccionar archivos por el procedimiento conocido como interleave.

Si, por ejemplo, se tiene que mostrar un texto en la pantalla mientras se reproduce un vídeo al tiempo que, como fondo, suena una música de acompañamiento, la aplicación tiene que trabajar con tres fuentes de datos de tres archivos diferentes y además en tiempo real.

2.7.2.2.1 El formato Lógico

La base de todo medio de almacenamiento de datos la constituye siempre el formato físico del soporte de datos. Además, si se quiere acceder a los datos almacenados no en forma de sectores sino como archivos y directorios, se precisa de un formato lógico.

Todo los CD-ROM que actualmente inserta en la unidad de su PC esta provistos de este formato.

2.7.2.2.2 Sectores Lógicos

Para no perderse en el nivel lógico de los sectores físico, el formato HSG define en primer lugar el sector lógico. Este, en cuanto a su tamaño, esta orientando a los sectores físicos según el Yellow Book¹ y contiene 2048 Bytes, es decir 2kB. Cada sector posee un número inequívoco, el denominado “logical sector number”, abreviado LSN. El primer LSN direccionable lleva el numero 0. Los primeros 150 sectores físicos que constituyen los dos primeros segundos de un CD no pueden direccionarse desde el nivel de formato lógico.

2.7.2.2.3 Bloques Lógicos

Para poder direccionar mejor los elementos de los sectores lógicos y al mismo tiempo refinar la glanulosidad de los mismos, HSG divide nuevamente el sector lógico en varios bloques lógicos. Cada bloque lógico (LBN) puede tener un tamaño de 512 bytes, 1024 bytes o 2048 bytes lo cual, en ultimo caso, se corresponde con el tamaño del sector lógico. Los LBN también se direccionan con números. Hay un 0 para el primer “bloque” lógico del primer sector “lógico, un 1 para el segundo, un 2 para el tercero y un 3 para el cuarto. El bloque lógico 4 se encuentra al principio del segundo sector “lógico”.

2.7.2.2.4 Path Table

Guardar los directorios como si de un archivo se tratase es un procedimiento sencillo, pero no exento de inconvenientes. En la búsqueda de archivos en subdirectorios de niveles dentro de la estructura, pues se tienen que buscar y leer demasiados archivos de directorios.

¹ Yellow Book...- Normas del Atapi.

Por este motivo y de la lenta velocidad de la búsqueda de las unidades de CD-ROM, se construye una especie de abreviación de los subdirectorios llamados Path Table. En el Path Table se enumeran los nombres de todos los directorios y subdirectorios de un CD. Si se tiene esta tabla de memoria, basta la lectura de un sector para averiguar la dirección de un archivo, siempre que, la entrada de directorio del archivo se encuentra en el primer sector de los datos de directorio. El Path Table contiene números de sectores enteros de 32 bits, sobre un CD-ROM hay dos copias de Path Table, una con números en formato Intel y otra en formato Motorola.

2.7.2.3 Extended attribute records (XAR)

Estos registros ofrecen la posibilidad de, al crear un archivo, almacenar cualquier información sobre el mismo.

Para que los archivos de directorio no aumenten con estas informaciones, estas constituyen el primer sector lógico del mismo. Cuando se accede a un archivo para lectura, el sistema operativo debe averiguar cuantos sectores XAR tiene para poder así saltarlos.

2.7.2.4 Técnicas de Grabación en CD

Uno de los factores que contribuyeron al éxito del CD fue la existencia de una normativa desarrollada conjuntamente por Sony y Philips gracias a la cual puede reproducirse cualquier disco. Esas normas fijan básicamente las siguientes variables:

- *Un formato de datos*, con indicaciones sobre el espacio para éstos, el direccionamiento de la información y los códigos de corrección de errores.
- Códigos básicos de canal y de corrección de errores (ECC), y espacio suficiente para más datos y ECC.
- Estructura microscópica y microscópica del disco.

Al comprobar las posibilidades como soporte de almacenamiento de datos del CD, Sony y Philips impulsaron también la normalización de un CD ROM. El disco, la técnica de exploración con láser y los métodos de grabación de masters y prensado de este CD ROM normalizado son idénticos a los correspondientes del CD. La diferencia, materializada en su totalidad respetando el formato de datos del CD, estriba en el uso de mecanismos más potentes de corrección de errores y en el direccionamiento absoluto más explícito de bloques de datos.

Recordemos que toda la información que usa la computadora es binaria (ceros y unos).

La información se graba en el disco maestro mediante un láser de potencia, cuyo haz se mueve radialmente sobre el disco mientras éste gira.

2.7.3 Buses de Datos

El BUS es el conjunto de líneas (cables) de hardware utilizados para la transmisión de datos entre los componentes de un sistema informático. Un bus es en esencia una ruta compartida que conecta diferentes partes del sistema como el procesador, la controladora de unidad de disco, la memoria y los puertos de entrada, salida, permitiéndoles transmitir información.

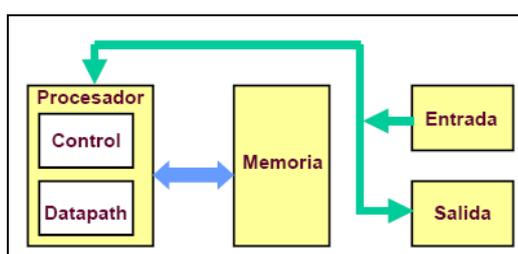


Figura 2.10 Esquema del bus de datos

Cuando un dato pasa de un componente a otro, viaja a lo largo de éste camino común para alcanzar su destino. Cuando un nuevo componente se inserta en uno de los conectores de expansión, queda unido directamente al bus, convirtiéndose en un objeto más de la unidad completa.

Cualquier información que entra o sale de un sistema ordenador se almacena temporalmente en al menos una de las distintas localizaciones que existen a lo largo del bus. La mayor parte de las veces el dato se sitúa en la memoria principal, que en la familia PC está formada por miles de posiciones de memoria de 8 bits. Pero algún dato puede acabar en un puerto, o registro, durante unos instantes, mientras espera que la CPU lo envíe a una posición adecuada.

Cuando el dato está listo para ser transferido, se transmite primero su dirección de destino por el bus de direcciones; el dato sigue a la zaga por el bus de datos.

Por tanto, el bus transporta algo más que datos. Lleva información de control, tales como las señales de temporización (del sistema reloj), las señales de interrupción, así como las direcciones de las miles de posiciones que forman tanto la memoria como los dispositivos que están conectados al bus. Para diferenciar estas cuatro funciones diferentes, el bus está dividido en cuatro partes: líneas de potencia, bus de control, bus de direcciones y bus de datos.

2.7.3.1 Organización de la comunicación del bus

2.7.3.1.1 Maestro/esclavo

Una transacción a través del bus incluye dos partes:

2.7.3.1.1.1 Petición de operación a través de un comando

El comando lleva asociada una dirección para indicar a qué dispositivo va dirigido

2.7.3.1.1.2 Transferencia del dato

El maestro es quién inicia la transacción en el bus mandando el comando (y la dirección si es necesario). El esclavo es el que responde al maestro enviando el dato al maestro o recibéndolo del maestro

En los casos de entrada/salida el procesador es el maestro (El comando es simplemente la dirección, R/W y CS).

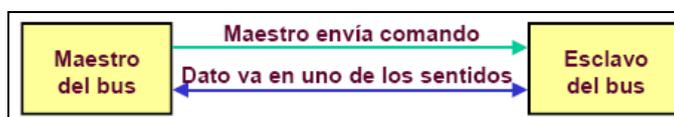


Figura 2.11 Utilización del Bus de Datos

2.8 SD CARD

Las tarjetas SD y Multimedia Card (y todas las demás tarjetas flash) se incluyen en el grupo de las memorias no volátiles, es decir en el grupo de memorias a las que no es necesario mantener alimentación para conservar la información.

La mayor capacidad, y el reducido coste han convertido a las memorias SD Card en el tipo de memoria no volátil más utilizado en dispositivos móviles. Estas se venden en forma de integrados, existiendo infinidad de modelos con diferentes prestaciones orientados al mercado industrial, y también en forma de tarjetas externas orientadas al mercado de consumo (cámaras, agendas electrónicas, teléfonos móviles).

Existen diferentes tipos y modelos de tarjetas Flash, todas con prestaciones similares en cuanto a capacidad, tamaño, y precio.

Actualmente los tipos más extendidos son:

- Compact Flash
- SD-MultiMediaCard (y sus versiones Mini)
- Smart Media
- Memory Stick
- XD.



Figura 2.12 Distintos tipos de tarjetas de memoria Flash disponibles en el mercado. De izquierda a derecha: MemoryStick (Sony), XD (Olympus Optical y Fuji Photo Film), SD/MultiMediaCard , SmartMedia (Olympus Optical y Fuji Foto Film) , Compact Flash y MiniSD.

2.8.1 Características de la SD CARD

2.8.1.1 Características generales

La comunicación de las tarjetas SD esta basada en una interfase de 9 pines: una línea de reloj (CLK), una de comandos (COMMAND), 4 líneas de datos y 3 líneas más de alimentación. Esta diseñada para operar en rangos bajos de voltaje. El protocolo de comunicación fue definido como parte de su especificación, pero también soporta operaciones MultiMediaCard. Además, también ofrece la posibilidad de usar el protocolo estándar SPI, lo cual la hace compatible con controladores ya existentes.

La tarjeta contiene chips de memoria flash diseñados especialmente para su uso como medio de almacenamiento. Además de esto, las tarjetas incluyen un controlador inteligente que administra diferentes protocolos de comunicación, algoritmos de seguridad para la protección contra copia no autorizada de la información almacenada, algoritmos de corrección de errores de código, manejo de defectos, diagnósticos, y administración de potencia.

2.8.1.2 Características físicas de la SD CARD

Las tarjetas son muy ligeras y pequeñas pesan cerca de 20g y miden cerca de 32mm de largo, 24mm de ancho y 2.1mm de grosor.

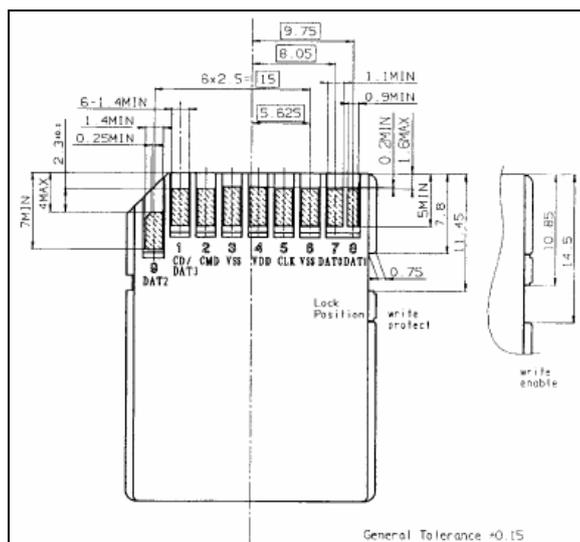


Figura 2.13 Características físicas de la tarjeta

2.8.1.3 Características eléctricas de la SD CARD

Las tarjetas trabajan en un rango de voltaje de 2.7 a 3.6 V y el consumo de corriente ($T_a=25^{\circ}\text{C}$ 3V) varía dependiendo del estado en que se encuentren y tiene como máximos los siguientes:

Estado	Valor	Unidades
Sleep	250	Ua
Read	65	Ma
Write	75	Ma

Tabla 2.1 Rangos de voltaje y corriente SD CARD

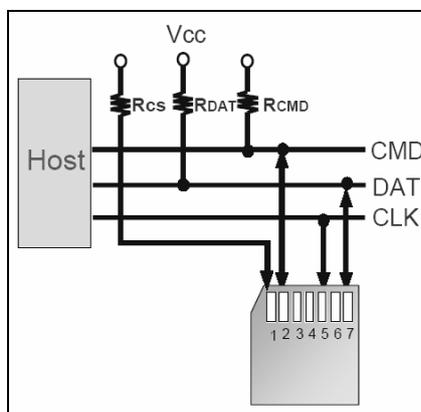


Figura 2.14 Características eléctricas de la tarjeta

Valores de las resistencias:

- RCS: 3.3 KΩ.
- R DAT: 3.3 KΩ.
- RCMD: 3.3 KΩ.

2.8.2 Modos de Operación

Las tarjetas con tecnología SD Card tienen dos modos de operación, el modo SPI y el modo SD. Las diferencias entre los modos involucran principalmente la función de cada uno de los pines, el protocolo usado y los comandos que son específicos de cada modo.

A continuación se detallan las características principales de ambos modos.

2.8.2.1 Modo SD

Este modo permite que se utilicen las terminales de datos (D0 - D3) en forma bidireccional, lo cual da mayor ancho de banda durante las transmisiones.

Los comandos se transmiten por la línea CMD en forma serial, y por último la respuesta de la tarjeta al comando se transmite por la línea CMD. La principal ventaja de éste modo.

Al inicio de la transferencia solo se envían datos por D0, ya después se pueden ampliar el ancho de banda de los datos hasta D3.

PIN	NOMBRE	TIPO	DESCRIPCION SD
1	CD/DAT32	I/O3	Card Detect/Data Line (Bit 3)
2	CMD	I/O	Command/Response
3	VSS1	S	Suply voltaje ground
4	VDD	S	Suply voltaje
5	CLK	I	Clock
6	VSS2	S	Suply voltaje 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 2.2 Aplicación de los Pines en el Modo SD

2.8.2.2 Modo SPI

En éste modo la tarjeta solo permite la entrada de datos por DATA IN (2) y la salida de datos por el terminal DATA OUT (7). Los comandos se mandan también por el terminal DATA IN.

La habilitación de la tarjeta se hace por la terminal CS (1). La señal de CLK que se envía desde el host o controlador es la que establece la velocidad de la comunicación, es por esto que el diseño del host se facilita ya que no se tienen problemas de desincronización cuando el host es más lento que la tarjeta. Sin embargo, como sólo se envían los datos por una terminal se tiene menor utilización del potencial de la memoria. Ya que el protocolo SPI es un estándar muy conocido, se decidió implementar este protocolo de comunicación con la tarjeta para la aplicación.

PIN	NOMBRE	TIPO	DESCRIPCION SD
1	CS	I	Chip Select
2	Cata in	I	Host to card commands and data
3	VSS1	S	Suply voltaje ground
4	VDD	S	Suply voltaje
5	CLK	I	Clock
6	VSS2	S	Suply voltaje ground
7	DataOut	O	Card to Host Data and Status
8	RSV(2)	I	Reserved
9	RSV(2)	I	Reserved

Tabla 2.3 Aplicación de los Pines en el Modo SPI

2.8.3 Organización de la Tarjeta

La organización interna de la tarjeta está constituida por un área de almacenamiento, una zona de registros de configuración/información acerca de la misma.

A su vez, el área de almacenamiento se subdivide en 2 partes independientes: el área protegida, la cual no se puede acceder a menos que el dispositivo se identifique, y en el área de almacenamiento común. Además, posee 5 registros de información, el CID, CSD, OCR, SCR y RCA1. Para los objetivos de este proyecto se desplegaron sólo los registros CID y CSD.

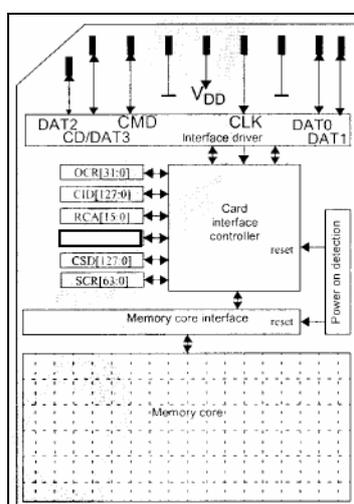


Figura 2.15 Organización física interna de la tarjeta

El registro CID (Card Identification Information) posee una longitud de 16 bytes y almacena un número único de identificación de la tarjeta.

El envío de este comando proveyó de varias características importantes acerca de una tarjeta en particular: la capacidad del dispositivo, y el formato mediante el cual son almacenados los datos en el mismo.

La tarjeta posee una organización de almacenamiento para área de datos similar a la de un disco duro.

FILE_FORMAT_GRP	FILE_FORMAT	TYPE
0	0	Hard disk-like file system with partition table
0	1	DOS FAT (floppy-like) with boot sector only (no partition table)
0	2	Universal File Format
0	3	Others/Unknown
1	0,1,2,3	Reserved

Tabla 2.4 Tipos de Formatos

Las tarjetas SD poseen un mecanismo de protección de Copyright que les permite identificarse cuando se accede a través de una PDA, o cualquier otro dispositivo.

Dicha información es almacenada en una partición de la tarjeta; ésta zona es conocida como Write Protection Group (WP Group).

Como se mencionó se encuentra en una partición aislada del área de almacenamiento y posee su propia tabla de particiones, así como comandos especiales para su acceso. Esta partición es de tamaño variable y viene especificada por el CSD de la tarjeta. La unidad básica de lectura de una tarjeta SD es un bloque de 512 bytes llamado sector. La estructura interna de la tarjeta está decidida en el sector cero de la misma el cual contiene el master boot record and partition table, la cual provee el bloque en el cual se encuentra la única partición de datos simples.

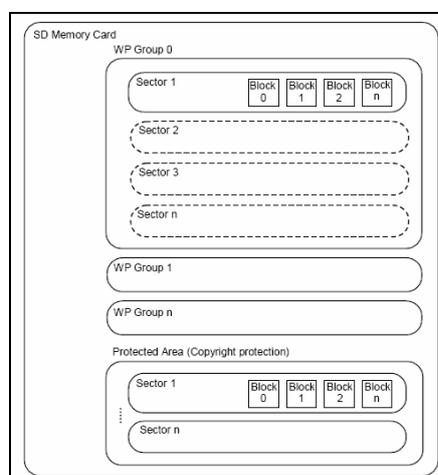


Figura 2.16 Organización del mapa de memoria de la tarjeta

Master Boot Record and partition table
Partition Boot Sector
FAT1
FAT2
Root Directory
Root Data

Tabla 2.5 Organización lógica interna de la tarjeta

2.9 MICROCONTROLADOR

Se denomina microcontrolador a un dispositivo programable capaz de realizar diferentes actividades que requieran del procesamiento de datos digitales como analógicos, de control y comunicación digital de diferentes dispositivos.

2.9.1 Microcontrolador PIC16F877A

En el desarrollo del prototipo se utilizó el PIC 16F877A. Este microcontrolador es fabricado por MicroChip² familia a la cual se le denomina PIC.

CARACTERÍSTICAS	16F877A
Frecuencia máxima	DX-20MHz
Memoria de programa flash palabra de 14 bits	8KB
Posiciones RAM de datos	368
Posiciones EEPROM de datos	256
Puertos E/S	A,B,C,D,E
Número de pines	40
Interrupciones	14
Timers	3
Módulos CCP	2
Comunicaciones Serie	MSSP, USART
Comunicaciones paralelo	PSP
Líneas de entrada de CAD de 10 bits	8
Juego de instrucciones	35 Instrucciones
Longitud de la instrucción	14 bits
Arquitectura	Harvard

² www.microchip.com

CPU	Risc
Canales Pwm	2
Pila Harware	-
Ejecución En 1 Ciclo Máquina	-

Tabla 2.6 Características del Microcontrolador PIC16F877A

Este microcontrolador posee las características requeridas para el desarrollo del prototipo planteado, las cuales se muestran a continuación.

- a. Puertos Seriales USART y SPI.
- b. Puertos de Comunicación de 8 bits.
- c. Memoria EEPROM.
- d. Conversor A/D.
- e. Contadores internos.

2.9.1.1 Programación

La programación de las sentencias que pondrá en funcionalidad el proyecto es realizado en lenguaje C, exclusivamente con un compilador CC5X³ conjuntamente con el MPLAB de la casa Microchip.

2.10 SENSORES



Figura 2.17 Sensor Inductivo

Los sensores inductivos hacen uso de las propiedades magnéticas de diversos materiales, y de las variaciones de diferentes parámetros asociados a los circuitos magnéticos (longitudes o secciones de núcleos, entrehierros, y otros),

³ www.bknd.com

para alterar la inductancia de bobinas normalmente fijas, consiguiendo variar la geometría del circuito magnético, permitiéndole detectar la presencia de objetos metálicos.

2.10.1 Interruptor Reed (Magnético):

Este es un interruptor normalmente abierto el cual al entrar en un campo magnético (rodearlo con Imán) cierra sus contactos y permite el funcionamiento del circuito al que este conectado, pero al salir de este campo vuelve a su posición normal, por lo cual se hace necesario hacer un circuito de retención con un relé. Este interruptor se conecta en serie al aparato que se quiera manejar (bombillo, sirena), su mayor aplicación es en los circuitos de alarmas residenciales.

El principio de funcionamiento de este interruptor se basa en que cierra sus contactos internos cuando se encuentra inmersa en un campo magnético.

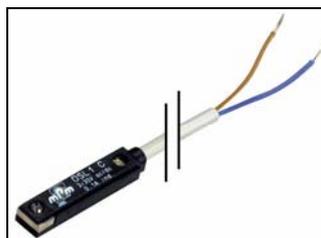


Figura 2.18 Sensor tipo REED

2.10.2 Montaje

En esta parte de montaje de los sensores inductivos se analiza una serie de características, relacionadas con el entorno de trabajo, que se deben tener en cuenta, tanto en el diseño como en el montaje, como son:

Las interferencias mutuas, los metales circundantes, los ruidos electromagnéticos y las asociaciones eléctricas de sensores, que pueden hacer que su funcionamiento no sea el deseado. Los sensores inductivos debido a su elevada conmutación del punto de trabajo se pueden emplear para determinar el sentido de giro y número de vueltas de un engranaje o eje.

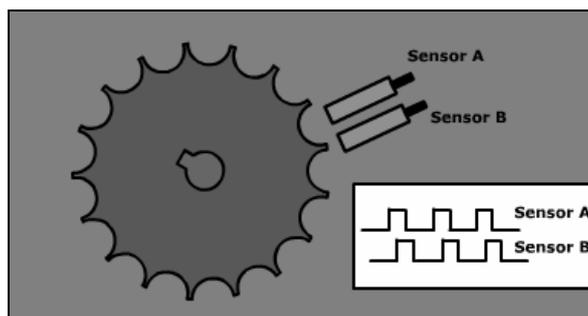


Figura 2.19 Sensores inductivos detección del número de vueltas y sentido de giro.

2.10.3 Protección

Las normas referentes a los tipos o grados de protección son:

2.10.3.1 NEMA

2.10.3.1.1 Clasificación NEMA

- **TIPO 1** Propósito general. Envolvente destinada a prevenir de contactos accidentales con los aparatos.
- **TIPO 2** Hermético a gotas. Previene contra contactos accidentales que pueden producirse por condensación de gotas o salpicaduras.
- **TIPO 3** Resistencia a la intemperie. Para instalación en el exterior.
- **TIPO 3R** Hermético a la lluvia.
- **TIPO 4** Hermético al agua. Protege contra chorro de agua.
- **TIPO 5** Hermético al polvo.
- **TIPO 6** Sumergible en condiciones especificadas de presión y tiempo.
- **TIPO 7** Para emplazamientos peligrosos Clase I. El circuito de ruptura de corriente actúa al aire.
- **TIPO 8** Para emplazamientos peligrosos Clase I. Los aparatos están sumergidos en aceite.
- **TIPO 9** Para emplazamientos peligroso Clase II y funcionamiento intermitente.
- **TIPO 10** A prueba de explosión.
- **TIPO 11** Resistente a ácidos o gases.

- **TIPO 12** Protección contra polvo, hilos, fibras, hojas, rebose de aceite sobrante o refrigerante.
- **TIPO 13** Protección contra polvo. Protege de contactos accidentales y de que su operación normal no se interfiera por la entrada de polvo

2.10.3.2 DIN

La norma DIN 40 050 establece la grado de protección **IP**; éste se compone de dos dígitos:

- El primero indica la protección contra sólidos.
- El segundo indica la protección contra el agua.

CUERPOS SÓLIDOS		AGUA	
0	No está protegido contra el ingreso de cuerpos extraños.	0	Sin protección.
1	Protegido contra ingreso de cuerpos extraños de hasta 50 mm de diámetro.	1	Protección contra el goteo de agua condensada.
2	Protegido contra ingreso de cuerpos extraños de hasta 12 mm de diámetro.	2	Protección contra el goteo hasta 15° de la vertical.
3	Protegido contra ingreso de cuerpos extraños de hasta 2.5 mm de diámetro.	3	Protección contra lluvia con ángulo inferior a 60°.
4	Protegido contra ingreso de cuerpos extraños de hasta 1 mm de diámetro.	4	Protección contra salpicaduras en cualquier dirección.
5	Protección contra depósito de polvo.	5	Protección contra el chorreo de agua en cualquier dirección.
6	Protección contra ingreso de polvo.	6	Protección contra ambientes propios de las cubiertas de los barcos.
		7	Protección contra la inmersión temporal.
		8	Protección contra la inmersión indefinida.

Tabla 2.7 Especificaciones DIN

2.11 GPS (Sistema de Posicionamiento Global)

El prototipo a implementar consta de la utilización de un GPS que nos dará la posición en un determinado punto, para lo cual se a utilizado el modulo receptor GPS ET-102 de GlobalSat⁴ dado que tiene las siguientes prestaciones:

- Exclusivo para proyectos de diseño.
- Comunicación serial Full duplex RS232.
- Manejo de protocolos NMEA.
- Control de Mensajes de Entrada y Salida.
- Precisión:
 - Velocidad 0.1 (m/s).
 - Posición 3 metros.
- Velocidad de transmisión 4800 bps.
- Compatible con TTL.
- Rango de medición:
 - Velocidad máxima: 515 (m/s).
 - Altitud: 18000 (metros).

2.11.1 Principio de Funcionamiento.

El GPS es un sistema que a través del uso de satélites, el conocimiento de su órbita y su ubicación real (mínimo 3 satélites, recomendables 4) determina la posición de un punto específico sobre el globo terráqueo, Consta de una constelación de satélites operando en órbitas de 12 horas a una altitud de 20.000 Km.

Esta compuesta de 24 satélites en 6 planos orbitales, cada uno igualmente separado del ecuador, e inclinado 55 grados. Con un segmento de control, que consiste en un centro de control maestro y un número de estaciones de monitoreo ampliamente separadas.

⁴ www.usglobalsat.com

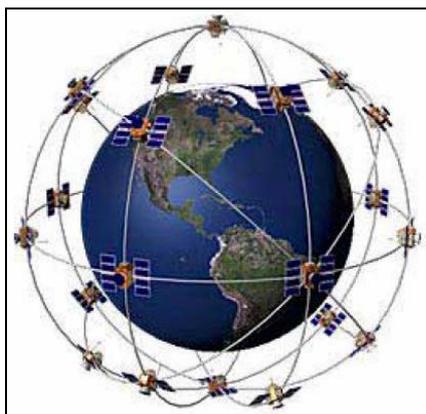


Figura 2.20 Posicionamiento de los Satélites GPS.

La red de control de tierra sigue los satélites, determina sus órbitas y periódicamente actualiza los efemeris, almanaques y otros datos del sistema. Para que luego estos se retransmitan al segmento de usuario. El segmento de usuario son todos los receptores GPS y su equipo de soporte.

La posición de los receptores es determinada por la distancia del satélite al receptor.

2.11.1.1 Mensaje de Navegación

El mensaje de navegación del GPS consiste en unas tramas temporizadas de datos que marcan el tiempo de transmisión de cada subparte del mensaje en el momento en que son transmitidas por el satélite.

Una trama de datos ocupa 1500 bits divididos en subpartes de 300 bits. Las tramas se transmiten cada 30 segundos (50 bit/s).

Tres subpartes de 6 segundos contienen datos orbitales y temporales. Las correcciones del reloj son transmitidas en la primera subparte y en la segunda y tercera se transmiten datos de información orbital (efemérides).

La cuarta contiene información del estado del satélite y otros parámetros de navegación y la quinta se usa para transmitir el almanaque de un satélite.

Cada 12 minutos y medio se retransmite toda la información completando 25 tramas completas.

Dichos datos son procesados por el MCC (Master Control Center).

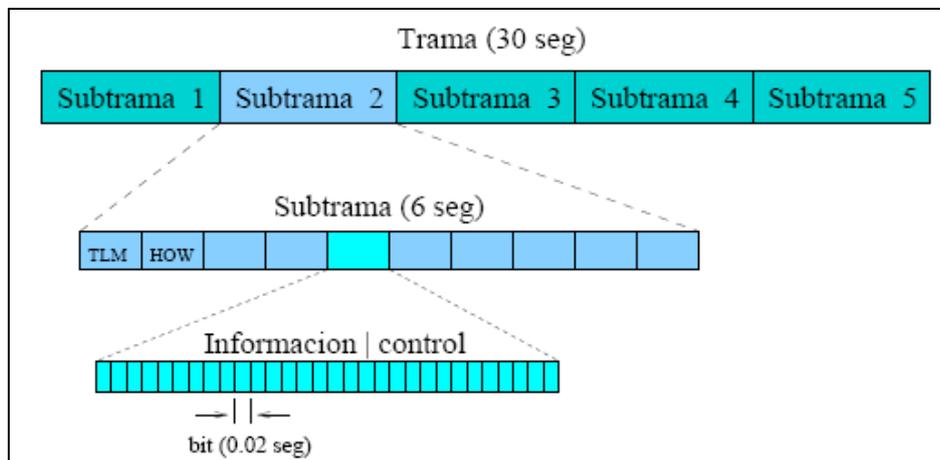


Figura 2.21 Mensajes de Navegación.

Trama Completa del GPS = 1500 bits	
Primera subtrama	Datos de telemetría (TLM): primeros 8 bits en subtrama. handover (HOW): que permite el cambio de C/A a Básicamente en esta subtrama se realizan las correcciones previas (reloj, atmósfera, etc). El final de todas las subtramas están compuestas por unos bits de paridad que corrigen datos finales.
Segunda subtrama	TLM, HOW, mas datos de efemérides (orbita exacta del satélite) + datos de paridad.
Tercera subtrama	TLM, HOW, mas datos de efemérides + datos de paridad. (espero que para la próxima pueda exponer más información sobre la segunda y tercera subtrama)
Cuarta subtrama	TLM, HOW. En esta trama se arma el dato de navegación una vez completada las 25 tramas (con 125 subtramas). Datos de paridad al final.

Quinta subtrama	<p>TLM, HOW.</p> <p>Almanaque (con este dato se clasifican los satélites mejores posicionados para realizar la transferencia de datos) para los satélites y estado de la constelación de los mismos.</p>
------------------------	--

Tabla 2.8 Trama del Sistema de Transmisión GPS.

Existen diferentes mensajes de salida que tiene el estándar **NEMA** con sus siglas “ **National Marine Electronics Association** ”, la cual regula las tramas de datos que debe recibir el usuario, en nuestro proyecto se ha utilizado el NEMA - 0183 la cual nos permite la interfaz del dispositivo GPS con instrumentos electrónicos y de desarrollo.

En la tabla se muestra los datos que se puede recibir de un sistema GPS.

GGA	Time, position and fix type data.
GLL	Latitude, longitude, UTC time of position fix and status.
GSA	GPS receiver operating mode, satellites used in the position solution and DOP values.
GSV	The number of GPS satellites in view satellite ID numbers, elevation, azimuth, and SNR values.
MSS	Signal-to-noise ratio, signal strength, frequency, and bit rate from a radio-beacon receiver.
RMC	Time, date, position, course and speed data.
VTG	Course and speed information relative to the ground.
ZDA	PPS timing message (synchronized to PPS).
150	OK to send message.

Tabla 2.9 Mensajes de Salida del Estándar NEMA.

Para el desarrollo del prototipo se tomo la trama RMC, ya que en ella contiene todos los datos de nuestro interés. La trama es la siguiente:

\$GPRMC,161229.487,A,3723.2475,N,12158.3416,W,0.13,309.62,120598, ,*10

Name	Example	Units	Description
Message ID	\$GPRMC		RMC protocol header
UTC Time	161229.487		hhmmss.sss
Status ¹	A		A=data valid or V=data not valid
Latitude	3723.2475		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	12158.3416		dddmm.mmmm
E/W Indicator	W		E=east or W=west
Speed Over Ground	0.13	knots	
Course Over Ground	309.62	degrees	True
Date	120598		ddmmyy
Magnetic Variation ²		degrees	E=east or W=west
<i>Mode</i>	<i>A</i>		<i>A=Autonomous, D=DGPS, E=DR</i>
Checksum	*10		
<CR> <LF>			End of message termination

Tabla 2.10 Detalles de la trama RMC.

2.11.2 Componentes del Sistema

2.11.2.1 Sector Espacial

Este sector lo forman los satélites de la constelación NAVSTAR (Navegación por satélite en tiempo y distancia). La constelación está formada por seis planos orbitales, y en cada uno de ellos existe una órbita elíptica casi circular donde se alojan los satélites regularmente distribuidos.

Los planos tienen una inclinación de 55° respecto al plano del ecuador, y se nombran como A, B, C, D, E y F. Cada órbita contiene al menos cuatro satélites, aunque pueden contener más. Los satélites se sitúan a una distancia de 20200 Km. respecto del geocentro, y completan una órbita en doce horas sidéreas. Estos satélites son puestos en funcionamiento por el Comando de las Fuerzas Aéreas Espaciales de U.S.A (AFSPC).

Con estos fundamentos, se garantiza la presencia de al menos cuatro satélites sobre el horizonte en todos los lugares de la superficie de la Tierra.

2.11.2.2 Sector de Control

El sector de control es el encargado de realizar un seguimiento de todos los satélites de la red NAVSTAR. Este sector se encarga específicamente de:

- Establecer la órbita de cada satélite, así como determinar el estado de sus osciladores.
- Con los parámetros anteriores obtenidos, se envía estos parámetros a los satélites para que éstos puedan difundirlos a los usuarios.

De este modo, el usuario recibe la información de las efemérides de posición de los satélites y el error que se está produciendo en su reloj, todo ello incluido en el mensaje de navegación.

Las Estaciones de Control de dichos satélites, son principalmente:

- Colorado Springs (U.S.A.). Central de cálculo y operaciones.
- Ascensión (Atlántico Sur).
- Hawai (Pacífico Oriental).
- Kwajalein (Pacífico Occidental).
- Diego García (Indico).

Existen además otras estaciones de seguimiento (láser, radar y ópticas), cuyo fin es la obtención de efemérides que no estén afectadas por la disponibilidad selectiva, denominadas precisas, y que están al alcance del usuario a través de organismos científicos como el IGS (International Geodynamic Service) o el NGS (National Geodetic Survey). Con ellas, tenemos la seguridad de posicionarnos en el sistema WGS84 con los errores típicos del sistema.

2.11.2.3 Sector de Usuarios

En este sector se considera al instrumental que deben utilizar los usuarios para la recepción, lectura, tratamiento y configuración de las señales, con el fin de

alcanzar los objetivos de su trabajo. Los elementos son el equipo de observación y el software de cálculo, que puede ser objeto de uso tras la campaña de observación, o bien realizable en tiempo real.

2.11.3 Equipo de Observación

Lo componen la antena, el sensor y la unidad de control o controlador.

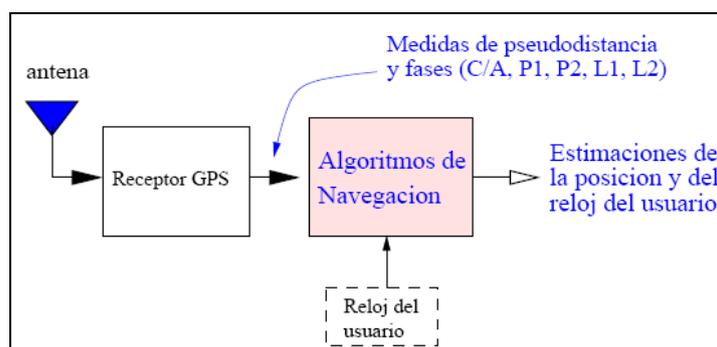


Figura 2.22 Componentes de un Sistema GPS.

2.11.3.1 La antena de recepción

Tiene la misión de recibir las radiaciones electromagnéticas que emiten los satélites y transformarlas en impulsos eléctricos, los cuales conservan la información modulada en las portadoras. Se denomina centro radioeléctrico de la antena al punto que se posiciona en nuestra observación. Dado que éste no suele coincidir con el centro físico, es conveniente orientar todas las antenas de una misma observación en la misma dirección con el fin de que el error se elimine.

2.11.3.2 El sensor

Recibe los impulsos de la antena receptora, y reconstruye e interpreta los componentes de la señal, es decir, las portadoras, los códigos y el mensaje de navegación. En definitiva, lo que hace es demodular la señal original.

Las técnicas de obtención de los *códigos* son, entre otras:

- **Correlación estrecha**. Se utiliza para reconstruir los códigos C/A y P, éste último cuando no está encriptado.
- **Correlación cruzada más cuadratura**. Esta técnica se utiliza para desencriptar el código P cuando el A/S está activado. También recibe esta técnica el nombre de “P-code adied”. Se fundamenta en encontrar el código W que es el responsable de que P no esté disponible para el usuario.
- **Z-Tracking™**. Es otra técnica para desencriptar el código P. Para ello, utiliza dos filtros de paso bajo para reducir el nivel de ruido. Hasta el momento, es la técnica que menor degradación produce en la señal (18 dB) y mejores resultados proporciona.

Para reconstruir las *portadoras* se utiliza principalmente:

- **Cuadratura**. Consiste en elevar la onda al cuadrado, limpiando dicha onda de toda la información modulada en ella (códigos y mensaje). Este método tiene el inconveniente de que se produce un empeoramiento importante de la relación señal/ruido por el aumento de éste último, produciendo una degradación importante en la señal.

Debemos añadir que toda señal recibida con una relación señal/ruido menor de 30 no debe ser considerada como útil en nuestros trabajos.

- **Reconstrucción a partir de los códigos**. Si por algunas de las técnicas anteriores hemos conseguido acceder a los códigos, podemos reconstruir las fases de las portadoras donde están modulados. La portadora L1 se puede reconstruir a través del código C/A y del código P, mientras que la L2 sólo a través del código P, ya que no contiene el C/A.

2.11.3.3 El controlador

Realiza las siguientes tareas:

- Controlar el sensor.
- Gestionar la observación.
- Almacenar los datos.

Es decir, las principales funciones del controlador son:

- Tipo de observación (estática, stop & go, cinemática, etc.).
- Parámetros de la observación (máscara de elevación, modo de grabación, determinación de las épocas, datos meteorológicos, etc.).
- Estado y salud de los satélites.
- Seguimiento de los mismos y calidad de la señal que transmiten.
- Filtrado de observaciones y datos.
- Definición y atributos de los puntos de observación.
- Estados de aviso en conceptos de geometría y pérdidas de ciclo.
- Definición del sistema de referencia.
- Tiempos de observación y actualización de tiempos.
- Control del nivel energético.
- Posición inicial y secuencial.

Los datos suelen ser grabados en unidades independientes de memoria o tarjetas RAM PCMCIA, que varían desde los 512 Kbytes a los 4 Mbytes de capacidad. También pueden ser almacenados directamente en un PC portátil conectado al receptor.

Es muy importante controlar la capacidad de grabación de datos y el tiempo de observación marcado. Tras la observación se obtienen los siguientes datos:

- Mensaje de navegación.
- Efemérides radiodifundidas por los satélites.
- Datos meteorológicos.
- Almanaque de estado de los satélites.
- Fichero de observación.

2.11.4 Errores y Aplicaciones futuras

2.11.4.1 Errores en GPS

El GPS no es perfecto y puede jugaros malas experiencias si no tenemos en cuenta que por ciertos factores, la información que nos muestra nuestro receptor puede ser no del todo correcta.

- Errores provocados por la Atmósfera: La naturaleza de la atmósfera está compuesta por muchas partículas que desgraciadamente atentan contra una medición exacta de nuestro GPS. Se destacan los errores por perturbación ionosférica, en donde se pueden apreciar errores en precisión de aproximadamente 10 metros.
- Errores en procesamiento de datos: el mal calculo de las efemérides, la microscópica pero segura imprecisión de los relojes atómicos en los satélites, y los errores en la posición de orbita, implican errores de cálculo que dan imprecisiones de 1 metro aproximadamente.
- Errores provocados por Disponibilidad Selectiva: Es un error impuesto a propósito por la DOD, el rango de error es de 70 metros aproximadamente.
- Error del receptor GPS: si nuestro recepto tiene alguna falla interna, por ende no nos dará información precisa.

Estos rangos de errores es lo que forma para los creadores de GPS el CEP (Error Circular Probable) el cual nos da un rango circular del dato en el receptor, con una probabilidad de acierto de un 95 %.

2.11.4.2 Aplicaciones Futuras

El mundo de GPS ha sido el componente más estable del sistema y ni siquiera, la futura puesta en marcha del sistema europeo GALILEO, parece inquietar a la importantísima industria desarrollada en torno al sistema NAVSTAR de los Estados Unidos. Es, en el componente hardware de GPS, donde se han producido y se producirán, importantes avances.

- Los servicios de transporte utilizan GPS para realizar un seguimiento de su flota y acelerar las entregas.
- Las compañías de transporte equipan los buques cisterna y cargueros con GPS para su navegación, así como para registrar y controlar los movimientos de las embarcaciones.
- Los pilotos civiles utilizan GPS para la navegación, fumigación aérea, topografía y fotografía aérea.
- Al utilizar la tecnología GPS para elaborar los planes de vuelo, las líneas aéreas ahorran millones de dólares. Los GPS se pueden utilizar para el aterrizaje instrumental, tanto en aeropuertos grandes como pequeños, y hacen posible la creación de nuevos sistemas de navegación aérea.
- En los automóviles se están instalando GPS para que los conductores puedan saber dónde están y a la vez recibir indicaciones de dirección. En Japón, 500.000 automóviles ya incorporan un sistema de navegación basado en GPS.

La tecnología GPS se esta desarrollando e implementando ya en ciertos sectores, como el del transporte frigorífico, ambulancias, transporte urbano, taxi,

grúas, etc., y es de esperar que se amplíe al resto de actuales sectores de usuarios, así como a los muchísimos nuevos campos de aplicación que están surgiendo día a día.

CAPITULO III

PROTOCOLOS DE COMUNICACIÓN Y SOFTWARE

3.1 INTRODUCCION

La comunicación de datos es el proceso de comunicar información en forma binaria entre dos o más puntos. El protocolo es un conjunto de reglas que posibilitan la transferencia de datos entre dos o más dispositivos. Y se requiere cuatro elementos básicos que son:

- Emisor: dispositivo que transmite los datos
- Mensaje: lo conforman los datos a ser transmitidos
- Medio: consiste en el recorrido de los datos desde el origen hasta su destino
- Receptor: dispositivo de destino de los datos

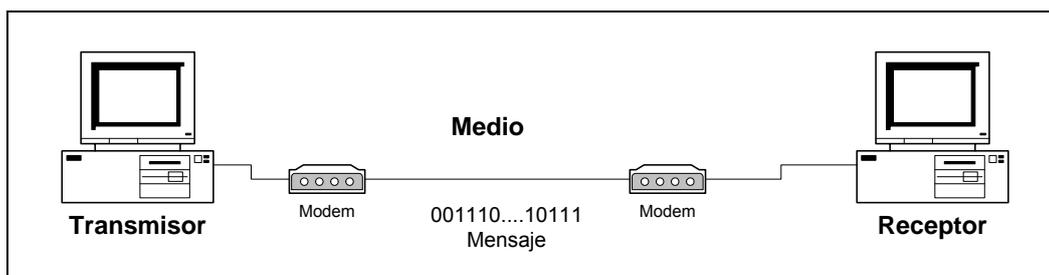


Figura 3.1 Comunicación de datos entre dispositivos

El prototipo de sistema de control de eventos utiliza protocolos de comunicación para la transmisión/ recepción en los modos SPI, ATAPI/ATA, NMEA y USART, los cuales serán descritos en el presente capítulo.

SPI (Serial Peripheral Interface), como su nombre lo indica es un tipo de comunicación serial. El protocolo esta basado en comandos, los cuales son enviados serialmente a través de DOUT y la respuesta al comando es recibida a través de DIN de una tarjeta SD card.



Figura 3.2 Transferencia de datos del protocolo SPI

Los comandos de la tarjeta tienen un tamaño fijo (6 bytes). Donde la convención es que primero se envía el bit más significativo del byte más significativo del comando. El formato de los comandos es el siguiente:

0	1	bit 5...bit 0	bit 31...bit 0	bit 6...bit 0	1
start bit	host	command	argument	CRC7 ¹	end bit

Figura 3.3 Estructura de comandos SPI

Donde el primer byte contiene el bit de inicio, el host y el identificador del comando a enviar, los siguientes 4 bytes contienen el argumento para dicho comando y el último byte representa el CRC7 (corrección de errores)

La USART (Universal Síncrono Asíncrono Recepción Transmisión), es *económica* porque utiliza pocas líneas de transmisión inclusive puede usar sólo una línea, es *confiable* porque los estándares actuales permiten transmitir datos con bits de paridad y a niveles de voltaje o corriente que los hacen poco sensibles a ruido externo.

Además por tratarse de información digital, los cambios en amplitud de las señales (normalmente causadas por ruido) afectan muy poco o nada a la información. Es *versátil* porque no está limitada a usar conductores eléctricos como medio de transmisión, pudiendo usarse también: fibra óptica, aire, vacío, etc. Además el tipo de energía utilizada puede ser diferente: luz visible, infrarroja, ultrasonido, pulsos eléctricos, radio frecuencia, microondas, etc.

NMEA (National Marine Electronics Association), la definición estándar de un NMEA-183 es una interfase eléctrica y un protocolo de datos para la comunicación entre instrumentos marinos. Este protocolo se lanza por primera vez en marzo de 1983, y su última versión fue publicada en el 2001.

El protocolo NMEA-183 es un protocolo estándar, prácticamente incorporado en todos los receptores GPS y admitido por la gran mayoría de los programas que permiten conexión a un GPS. Precisamente esta estandarización y su amplia difusión es la cualidad más destacable de este protocolo.

ATAPI/IDE (AT Attachment with Packet Interface/ Integrated driver Electronics). El protocolo ATA/ATAPI se ha desarrollado hace unos 20 años atrás, empezando originalmente con los discos duros de las PC AT, y actualmente están normadas por el comité ANSI T10.

Algunos de los pasos más importantes de este protocolo, se han reflejado en los estándares que han sido desarrollados durante este tiempo.

ATA-1: IDE (Intelligent Drive Electronics or Integrated Drive Electronics). El primer Standard ATA fue diseñado para soportar dos dispositivos en un mismo cable, una interfaz de 16-bit, y 3 modos PIO.

ATA-2: La segunda versión de ATA Standard fue diseñada para una rápida transferencia de datos de modo PIO, transferencia multiword DMA, y modo Logical Block Addressing (LBA). LBA fue una forma de trasladar las menciones de cylinder, head y sector mapping en una forma lineal de logical blocks.

También el ATAPI (AT Attachment Packet Interface) Standard fue escrito como una extensión de ATA-2 standard y para los CD-ROM.

ATA-3: Tiene las mismas características de las anteriores, pero es para modo faster DMA y modo PIO.

Los dispositivos ATAPI incluyen CD-ROMs, CD-Recordable, CD-RW, DVD, tape drives, floppy drives (ZIP and LS-120), y cargadores. Los controladores de dispositivos ATAPI utilizan el SCSI command packets y MMC command, que normalmente son comandos enviados serialmente, pero se han adaptado para el protocolo ATAPI/IDE, que es un protocolo paralelo.

IDE conector, la parte física de esta interfaz es el conector IDE, que son exclusivos para dispositivos ATAPI, y consta de 40 pines, que en el presente capítulo ampliaremos en su configuración.

3.2 PROTOCOLOS DE COMUNICACIÓN

3.2.1 Serial Peripheral Interface (SPI)

Este protocolo se utiliza para el acceso a las tarjetas SD CARD (Security Digital Card) y MMC (Multi Media Card) en modo SPI, y consiste en el intercambio de información entre el controlador (master) y la tarjeta (slave). Este intercambio se lleva a cabo mediante el envío de comandos por parte del controlador y de respuestas por parte de la tarjeta. Así, en la lectura, el controlador envía el comando de petición de lectura a la tarjeta y esta le envía la respuesta de confirmación seguida del bloque de datos con la información contenida a partir de la dirección solicitada. En la escritura el proceso es parecido, el controlador indica a la tarjeta mediante el comando de escritura que quiere escribir información en una determinada dirección, esta le responde indicando que esta lista y a continuación el controlador envía el bloque de datos a escribir. Las operaciones que no requieren intercambio de datos funcionan de igual forma pero sin usar bloques de datos.

3.2.1.1 Estructura del protocolo SPI

El intercambio de información se da a través de comandos que se envía desde el controlador hacia la tarjeta, y tienen un tamaño fijo (6 bytes). Donde la convención es que primero se envía el bit más significativo del byte más significativo del comando. La estructura es la siguiente:

Byte 1			Byte 2-5	Byte 6	
7	6	5-0	31-0	7-1	0
0	1	Command	Command Argument	CRC	1
Star bit	Host				End bit

Tabla 3.1 Transferencia de datos del protocolo SPI

La tarjeta identifica los comandos porque el primer byte de estos (byte 1) siempre comienza por 01, el resto de bits del primer byte contiene el número de comando codificado en binario natural.

Así el primer byte correspondiente al comando 0 (CMD0) sería: 01000000, o el primer byte correspondiente al comando 39 (CMD39) sería: 01100111.

Los siguientes 4 bytes (bytes 2 –5) contienen los argumentos del comando. En los comandos que no requieren argumentos estos valen 0.

El último byte (byte 6) es el byte de CRC para la verificación de errores y en realidad en el protocolo SPI no se utiliza, a no ser que en el registro de configuración se especifique que se desea utilizar CRC, y las respuestas de la tarjeta son bloques formados por 1 o 2 bytes, dependiendo del tipo de respuesta que se trate.

El tipo de respuesta es función del comando, es decir que cada comando tiene asociado un tipo de respuesta.

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

Tabla 3.2 Principales comandos del protocolo de acceso mediante SPI

Bit	Description
0	Idle State
1	Erase Reset
2	Illegal Command
3	Com CRC Error
4	Erase_Seq_Error
5	Address Error
6	Parameter Error
7	0

Tabla 3.3 Respuestas R1 y RB

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

Tabla 3.4 Respuesta R2

3.2.1.2 Modo de Transmisión y recepción de información.

Toda la comunicación entre la tarjeta y el controlador se realiza según el orden del esquema, de izquierda a derecha, es decir que primero se transmite el bit de mas peso (bit 7) del byte 1, y por último el bit de menos peso (bit 0) del byte 6, es decir es una transferencia More Significant Bit First (MSB First).

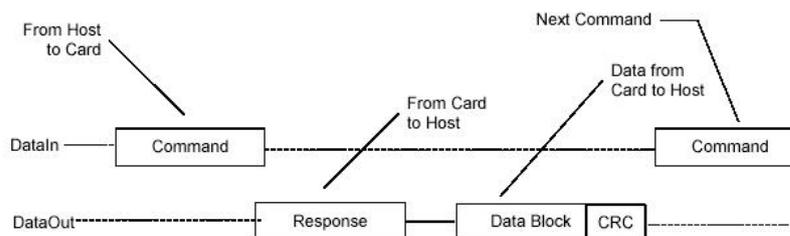


Figura 3.4 Lectura de un bloque de la tarjeta

En el caso de la **lectura** por ejemplo, al enviar el comando de lectura de un bloque (CMD 17) este debe llevar como argumento la dirección del bloque de 512 bytes que se desea. Es decir, que por ejemplo para leer el sector 2 de la tarjeta, el comando 17 debe ser modificado de la siguiente forma:

CMD17 5100 0002 0001H

Una vez que el comando sea recibido por la tarjeta, esta regresa una respuesta de 3 bytes y un token de inicio, seguidos del bloque de 512 bytes y el CRC del mismo. Este token es de la forma: **FF FF 00 FF FE**

Por otro lado, en el caso de la **escritura**, primero se envía el comando que de igual forma que en el caso de la lectura lleva como argumento la dirección del sector que se escribirá. Sin embargo, a diferencia de la lectura después de recibir la respuesta estándar de 3 bytes que indica que el comando se acepta, el host enviara un token de inicio seguido del bloque de 512 bytes de datos y dos token de fin de datos. Por último, una vez que la transmisión del bloque termina la tarjeta envía una respuesta que incluye si hubo error y/o si la tarjeta esta ocupada.

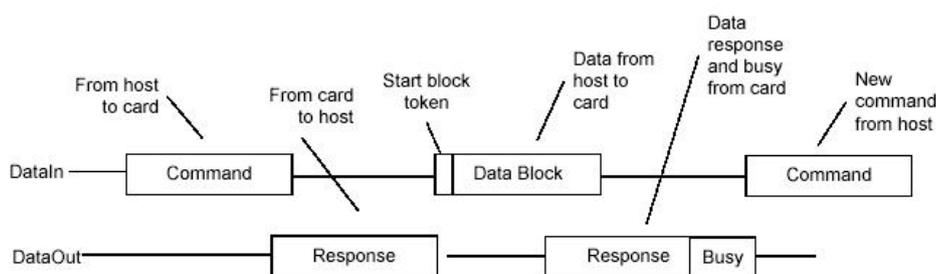


Figura 3.5 Escritura de un bloque a la tarjeta.

3.2.1.3 Configuración e inicialización en modo SPI de la SD Card y del PIC 16F877A.

SD CARD en modo SPI

Reset: Por defecto, al arrancar la tarjeta, esta se encuentra en modo MultiMediaCard. Para que entre en modo SPI, hay que enviarle el comando 0

mientras se mantiene activa la señal \bar{CS} ($\bar{CS}=0$), pero antes de todo, para poder iniciar la comunicación por el bus hay que enviar como mínimo 74 ciclos de clock a la tarjeta. Y se debe seguir la siguiente secuencia:

- Dar como mínimo 74 ciclos de clock, es decir enviar unos 10 bytes.
- Activar la señal \bar{CS} ($\bar{CS}=0$).
- Enviar el comando 0 con el CRC bien calculado, ya que todavía no estamos en modo SPI, por lo que sí se considera el CRC. De hecho la secuencia del comando 0 siempre es la misma: 0x40,0x00,0x00,0x00,0x00,0x95.
- Esperar el byte de respuesta que ha de ser 00000001 (tarjeta en modo idle).

Inicialización: Una vez reseteada y en modo SPI, hay que hacer la inicialización de la tarjeta, para ello hay que enviar el comando 1. La secuencia general es esta:

- Activar el \bar{CS} ($\bar{CS}=0$).
- Enviar el comando 1: 0x41,0x00,0x00,0x00,0x00, 0xXX. Como la tarjeta ya esta en modo SPI el CRC puede tomar cualquier valor.
- Esperar el byte de respuesta que ha de ser 00000000 (tarjeta lista).

PIC 16F877A en modo SPI

Una de las características importantes de este PIC es la implementación de pines exclusivos para el trabajo con este protocolo. Este modo permite la transferencia de datos de 8 bits en serie, que pueden ser transmitidos y recibidos de forma síncrona y simultánea. Para la comunicación se utilizan tres hilos de conexión:

- *Serial Data Out* (SDO): Salida de datos Serie (RC5)
- *Serial Data In* (SDI); Entrada de datos Serie (RC4)
- *Serial Clock* (SCK), Reloj de Sincronización (RC3)

Al inicializarse el módulo **SPI**, varias especificaciones necesitan ser configuradas. Esto se en los registros **SSPSTAT** y **SSPCON** (**SSPCON**<5:0> y **SSPSTAT**<7:6>). Pudiéndose seleccionar las distintas modalidades de trabajo:

- Modo Master (donde la señal de reloj SCK será de salida)
- Modo Esclavo (donde la señal de reloj SCK será de entrada)
- Polaridad del Reloj: (Cambio de estado por flanco ascendente o descendente de SCK, y su estado de inactivad)
- Fase de datos de entrada (Debemos de precisar el momento en el que se toma la muestra del dato de entrada, que puede ser en la mitad o al final del impulso)
- Selección del flanco del reloj (TX por flanco de bajada o subida de SCK)
- Velocidad de SCK (Solo en modo Master)
- Selección del modo Esclavo, si trabajamos de esta forma.

Los datos recibidos se introducen en serie en el registro de desplazamiento **SSPSR** que, una vez haya recibido el octavo bit, pasarán al registro **SSPBUF** en paralelo, poniéndose en ese momento los bits **SSPSTAT** <0> **BF** y **PIR1** <3> (**SSPIF**) a uno, indicando el primero de ellos que el buffer está lleno y el segundo para indicar que la recepción/transmisión ha finalizado.

Si durante la recepción/ transmisión se intenta escribir en el registro **SSPBUF**, la escritura será ignorada y el bit **WCOL** (**SSPCON**<7>) se pone a uno para indicar que ha habido “colisión”. El bit **BF** se pone a cero, cuando se proceda a leer el registro **SSPBUF**, lo cual debe realizarse ante de que tenga lugar una nueva recepción.

El registro **SSPSR** no se puede leer o escribir directamente, y solo se puede acceder a él a través del registro **SSPBUF**. Por lo tanto, por el registro **SSPBUF** deben pasar forzosamente todos los datos tanto en transmisión como en recepción.

3.2.1.4 Lectura y escritura de datos en la memoria SD Card

Lectura de un bloque de datos. Para leer un bloque de la tarjeta hay que enviar a esta el *comando 17* con la dirección de inicio de lectura en los bytes de argumento. La dirección puede tomar cualquier valor comprendido dentro del rango de direcciones válidas de la tarjeta pero todo el bloque leído debe estar dentro de un mismo sector físico. A continuación la tarjeta envía un byte de respuesta R1 seguido del bloque de datos, que comienza por 0xFE, continua con los bytes de datos y finaliza con los 2 bytes de CRC que no se usan. La secuencia a seguir es la siguiente:

- Activar el PIN \bar{CS} ($\bar{CS}=0$).
- Enviar el comando 17 0x51,0xXX,0xXX,0xXX,0xXX,0xYY. Los 4 bytes XX corresponden a la dirección a partir de la cual se quieren leer los datos. 0xYY corresponde al byte de CRC y como la tarjeta esta en modo SPI puede tomar cualquier valor ya que no se considera.
- Si todo va bien, la tarjeta responde con un byte de respuesta R1, seguido del bloque de datos con la información solicitada y que el controlador tendrá que ir capturando. Esta tiene la misma estructura utilizados en la escritura:
 - 1 byte de inicio de bloque de datos 0xFE
 - n bytes con los datos a guardar.
 - 2 bytes de CRC.
- Si se produce un error durante la comunicación, la tarjeta no transmitirá ningún dato y en lugar de estos enviará un byte indicador de error.

Escritura de un bloque de datos. Una vez inicializada la tarjeta, para escribir un bloque en esta, hay que enviar el comando 24 con la dirección de inicio a partir de la cual se desean guardar los datos. Si todo va bien la tarjeta enviará tres respuestas R1 repetidas informando al controlador que ya puede enviar el bloque de datos, que ha de tener una longitud de 512 bytes (en la escritura solo se permiten 512 bytes) más el byte de inicio de bloque de datos y los dos bytes de CRC. La secuencia a seguir es:

- Activar el PIN_iCS ($iCS=0$).
- Enviar el comando 24 0x58, 0xXX,0xXX,0xXX,0xXX,0xYY. Los 4 bytes XX corresponden a la dirección a partir de la cual se quieren guardar los datos. 0xYY corresponde al byte de CRC que toma cualquier valor.
- Si todo va bien la tarjeta responde con el byte de respuesta R1 tres veces consecutivas.
- Enviar a la tarjeta el bloque de datos que consiste en:
 - 1 byte de inicio de bloque de datos 0xFE
 - 512 bytes con los datos a guardar.
 - 2 bytes de CRC
- Mientras la tarjeta esta ocupada guardando el valor, irá enviando bytes indicando que está ocupada, y cuando finalice la escritura enviará un byte de confirmación.

3.2.2 Universal Synchronous Asynchronous Receiver Transmitter (USART)

El módulo transmisor (USART) es otro de los dos puertos serie de los que dispone esta familia de microcontroladores. Los módulos de I/O. (USART también se conocen como *Serial Communications Interface* o SCI) puede configurarse como una unidad de comunicación en serie para la transmisión de datos asíncrona con dispositivos tales como terminales de computadora o computadoras personales, o bien para comunicación síncrona con dispositivos tales como convertidores A/D o D/A, circuitos integrados o memorias EEPROM con comunicación serie, etc.

3.2.2.1 Protocolo de comunicación serie.

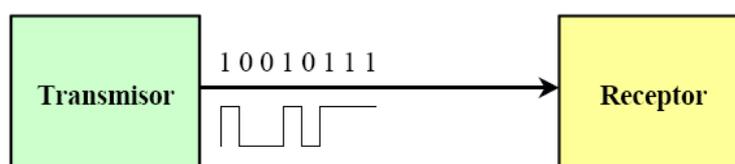


Figura 3.6 Transmisión serial

A diferencia de la comunicación en paralelo, en la comunicación en serie se hace necesario establecer métodos de sincronización para evitar la interpretación errónea de los datos transmitidos.

Para ilustrar esto consideremos la siguiente información en serie:
0100110001001100100

Esta información puede interpretarse de diversas maneras, (aún si se recibe a la velocidad adecuada) dependiendo del punto de inicio de separación de datos, por ejemplo, una posible interpretación sería como sigue: **0100110001 00110010 0**, que interpretado como códigos ASCII corresponde a los caracteres '1' y '2'. Sin embargo, otra posible interpretación es: **01001100010 01100100**, que corresponde a los caracteres 'b' y 'd'.

Sincronización de bit. Una manera de resolver el problema anterior es la sincronización de bits que puede realizarse por varios métodos:

- Enviar por una línea adicional una señal de reloj que indique el centro de las celdas de bits en la línea de datos (**datos no - auto reloj**).
- Enviar con cada bit y por la misma línea de datos información que permita extraer la señal de reloj (**datos auto reloj**).
- Lograr mediante alguna estrategia que los relojes de transmisión y de recepción se mantengan en fase continuamente.

Codificación no auto reloj. En la figura siguiente se muestran las tres codificaciones de una línea de datos:

RZ.- Una celda de bit es 1 si contiene un impulso positivo y un 0 si no lo contiene.

NRZ.- La celda contiene un 1 o 0 de acuerdo al nivel de la señal (const.) en la celda.

NRZI.- La celda de bit contiene un 1 si hay una transición y un 0 si no la hay.

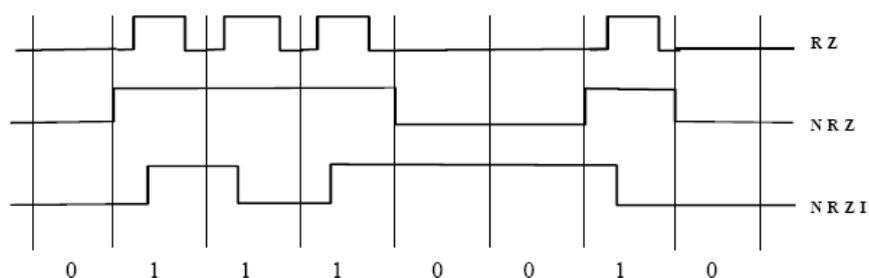


Figura 3.7 Codificación no auto reloj

Sincronización de carácter. Algunos sistemas utilizan líneas adicionales que envían impulsos para indicar el inicio de un bloque de caracteres. Otros sistemas que no requieren líneas adicionales a la línea de datos son:

Método Asíncrono. Cada carácter va señalizado mediante dos bits: *un bit de inicio* y *un bit de paro*, estos dos bits permiten al receptor reconocer el inicio y el final de cada carácter.

La especificación RS404 de EIA (Electronic Industries Association) define las características del método asíncrono para transmisión en serie de acuerdo a las siguientes reglas:

- Cuando no se envían datos la línea debe mantenerse en estado 1.
- Cuando se va a mandar un carácter se envía primero un bit de inicio de valor 0.
- A continuación se envían todos los bits del carácter a transmitir al ritmo marcado por el reloj de transmisión.
- Después del último bit del carácter enviado se envía un bit de paro de valor 1.

Método Síncrono. Cada mensaje o bloque de transmisión va precedido de unos caracteres de sincronismo. Así, cuando el receptor identifica una configuración de bits igual a la de los caracteres de sincronismo da por detectado el inicio y el tamaño de los datos.

3.2.2.2 Configuración de la USART del PIC16F877

Para el usuario de un microcontrolador que posee una USART o sistema similar la manera detallada como el sistema logra establecer la comunicación resulta transparente a él, ya que sólo tiene que configurar el protocolo del transmisor y del receptor para que estos logren la comunicación adecuada, es decir, el usuario usualmente sólo debe configurar:

- Tipo de comunicación (síncrona o asíncrona)
- Velocidad de transmisión en Baudios (bits por segundo)
- Longitud de los datos
- Bits de inicio y de paro, bits de paridad, etc.

La USART del PIC 16F877A puede ser configurada para operar en tres modos:

- Modo Asíncrono (full duplex (TX y RX simultáneas))
- Modo Síncrono – Maestro (half duplex)
- Modo Síncrono – Esclavo (half duplex)

Modo Asíncrono. En este modo la USART usa un formato estándar **NRZ** asíncrono, el cual para la sincronización usa: 1 bit de inicio (**I**), 8 o 9 bits de datos y 1 bit de paro (**P**). Mientras no se están transmitiendo datos la USART envía continuamente un bit de marca.

El modo asíncrono se selecciona limpiando el bit SYNC del registro **TXSTA (98H)**. El modo asíncrono es deshabilitado durante el modo SLEEP.

Cada dato es transmitido y recibido comenzando por el LSB. El hardware no maneja bit de Paridad, pero el noveno bit puede ser usado para este fin y manejado por software.



Figura 3.8 Estructura de la trama serial del PIC16F877A

El módulo Asíncrono de la USART consta de 4 módulos fundamentales:

- 1) El circuito de muestreo
- 2) El generador de frecuencia de transmisión (Baud Rate)
- 3) El transmisor asíncrono
- 4) El receptor asíncrono.

El circuito de muestreo. El dato en la patita de recepción (RC7/RX/DT) es muestreado tres veces para poder decidir mediante un circuito de mayoría, si se trata de un nivel alto o un nivel bajo.

El Generador de Baud Rate (BRG). Este generador sirve tanto para el modo síncrono como el asíncrono y consiste de un contador/divisor de frecuencia de 8 bits controlado por el registro **SPBRG (99H)**. De tal manera que la frecuencia de transmisión se calcula de acuerdo a la siguiente tabla:

SYNC	BRGH=0 (baja velocidad)	BRGH=1 (Alta velocidad)
0 (modo asíncrono)	Baud rate= $F_{osc}/(64(X+1))$	Baud rate= $F_{osc}/(16(X+1))$
1 (modo síncrono)	Baud rate= $F_{osc}/(4(X+1))$	-

Tabla 3.5 Generación de Baudios

En esta tabla X=valor de 8 bits en el registro del divisor, SPBRG. El bit BRGH corresponde a TXSTA<2>.

El transmisor asíncrono. Para transmitir un dato, el programa deberá ponerlo primero en el registro TXREG. En cuanto el TSR termina de enviar el dato que tenía (en cuanto transmite el bit de paro) lee el dato contenido en TXREG (si hay alguno). En cuanto el dato de TXREG es transferido al TSR el TXREG queda vacío esta condición es indicada mediante el bit bandera TXIF, el cual se pone en alto. Este bit no puede ser limpiado por software, sólo dura un instante en bajo cuando se escribe un nuevo dato a TXREG. Si se escribe un dato seguido de otro (back to back) a TXREG el primero se transfiere inmediatamente a TSR y el otro tiene que esperar hasta que el TSR termine

de enviar el bit de Stop del primero. Durante esta espera TXIF permanece en bajo.

Para habilitar el módulo de transmisión es necesario poner en alto el bit TXEN (TXSTA<5>), mientras no se habilite el módulo, la patita de transmisión (RC6/TX/CK) se mantiene en alta impedancia. Si TXEN es deshabilitada a la mitad de una transmisión, ésta será abortada y el transmisor será reseteado. De acuerdo a lo anterior, la inicialización del módulo de transmisión consiste en los siguientes pasos:

1. Inicializar baud rate escribiendo al registro SPBRG el divisor adecuado y opcionalmente al bit BRGH.
2. Habilitar comunicación asíncrona limpiando el bit SYNC y poniendo el bit SPEN.
3. Si se van a usar interrupciones, poner el bit TXIE (PIE<4>).
4. Poner el bit TX9 si se desea transmitir datos de 9 bits
5. Habilitar transmisión poniendo el bit TXEN, lo cual pondrá el bit TXIF.
6. Colocar el noveno bit del dato en TX9D si se están usando datos de 9 bits.
7. Cargar el dato al registro TXREG (inicia la transmisión).

El receptor asíncrono. El corazón del receptor es el registro de corrimiento RSR. Este registro no es accesible por software, pero, cuando el dato recibido se ha completado (se ha recibido el bit de Stop) el dato de RSR es transferido automáticamente al registro **RCREG (1Ah)** si éste está vacío y al mismo tiempo es puesto en alto la bandera de recepción RCIF (PIR1<5>).

La única manera de limpiar la bandera RCIF es leyendo el los datos del registro RCREG.

El registro RCREG puede contener hasta dos datos, ya que es un buffer doble que funciona como una cola de dos posiciones.

De acuerdo a lo anterior, la inicialización del módulo de recepción es como sigue:

1. Inicializar el baud rate escribiendo al registro SPBRG el divisor adecuado y opcionalmente al bit BRGH
2. Habilitar el puerto serie asíncrono limpiando el bit SYNC y poniendo el bit SPEN.
3. Si se van a usar interrupciones, poner el bit RCIE (PIE<5>).
4. Si se desea recepción de datos de 9 bits se deberá poner el bit RX9 (RCSTA<0>).
5. Habilitar la recepción poniendo el bit CREN (RCSTA<4>)
6. El bit RCIF se pondrá cuando la recepción de un dato se complete y se generará una interrupción si RCIE está puesto.
7. Leer el registro RCSTA para obtener el noveno bit (si se están recibiendo datos de 9 bits) o para determinar si ha ocurrido un error de recepción.
8. Leer los 8 bits del dato recibido leyendo el registro RCREG.
9. Si ocurrió algún error este se limpia al limpiar el bit CREN, el cual deberá volver a ponerse si se desea continuar la recepción.

3.2.3 Attachment with Packet Interface (ATAPI/IDE)

3.2.3.1 ATAPI Protocolo

Un dispositivo ATAPI es controlado por dos métodos originalmente por comandos ATA utilizando tareas de archivo, y un método de paquete de comandos (packet command), Los dos métodos usan la interfase IDE para ser programados por el computador.

Los comandos son enviados en forma paralela a todos los periféricos tanto para su ejecución como para la selección del dispositivo (DVR (0)). En la siguiente figura se muestra la estructura de una comunicación ATAPI.

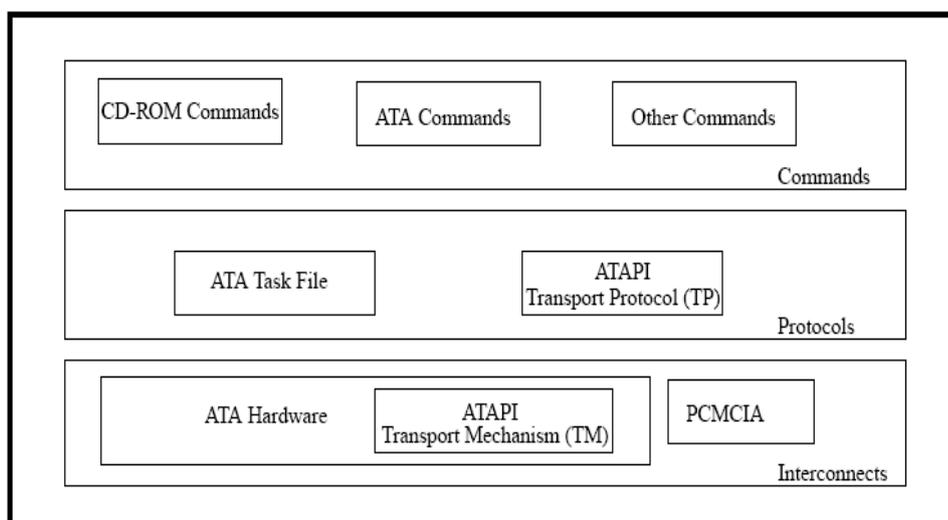


Figura 3.9 Estándar ATAPI/ATA

El Protocolo ATAPI se centra alrededor de la utilización del nuevo paquete de comandos llamado “ATAPI Packet Command.” La inicialización del dispositivo da únicamente con el reset, sea este a nivel de hardware o software.

3.2.3.2 Registros del un dispositivo ATAPI

Los registros que son controlados por BSY y DRQ son los siguientes, que a excepción del registro de datos que es de 16 bits (1 word), todos son de 8 bits.

Todo el funcionamiento e instrucciones para el dispositivo ATAPI se realiza mediante una comunicación de forma paralelo, a través del conector físico IDE.

Los registros son seleccionados a través de direcciones que cada uno posee, las cuales son con designadas mediante los pines DA0, DA1, DA2, y para la habilitación de los registros ya sean de control o de comandos se las realiza mediante los pines CS1FX y CS3FX. Hay que tomar en cuenta si los registros son de lectura o escritura para su respectiva habilitación de DIOR- y DIOW-⁵

⁵ Las variables con el signo (-), se refiere a la activación en bajo.

Addresses					Functions	
CS1FX	CS3FX	DA2	DA1	DA0	Read (DIOR-)	Write (DIOW-)
					Control Block Registers	
N	A	0	0	0	Floppy A Status	Unused
N	A	0	0	1	Floppy B Status	Unused
N	A	0	1	0	Unused	Floppy Digital Output Register
N	A	0	1	1	Floppy ID / Tape Control	RESERVED
N	A	1	0	0	Floppy Controller Status	RESERVED
N	A	1	0	1	Floppy Data Register	
N	A	1	1	0	Alternate ATAPI Status	Device Control
N	A	1	1	1	<i>Note¹</i>	Not Used
					Command Block Registers	
A	N	0	0	0	Data	
A	N	0	0	1	ATAPI Error Register	ATAPI Features
A	N	0	1	0	ATAPI Interrupt Reason Register	Unused
A	N	0	1	1	Reserved For SAM TAG Byte	
A	N	1	0	0	ATAPI Byte Count Register (bits 0-7)	
A	N	1	0	1	ATAPI Byte Count Register (bits 8-15)	
A	N	1	1	0	Drive Select	
A	N	1	1	1	ATAPI Status	ATA Command

Tabla 3.6 Registros de un dispositivo ATAPI

Registro de estado (Status). Es solo de lectura, pero cuando se escribe en el se convierte en el registro de comandos. Este registro nos permite en cualquier momento obtener el estado del dispositivo, en el envío, ejecución y resultados de los comandos.

D7	D6	D5	D4	D3	D2	D1	D0
BSY	DRDY	DMA READY or DF	SERVICE or DSC	DRQ	CORR	Reserved	CHECK

Tabla 3.7 Registro Status

- Bit 7 BSY, Maneja el acceso al bloque de comandos (Command Block).
- Bit 6 DRDY Indica que el dispositivo esta listo para responde a un ATA command.
- Bit 5 DMA READY / DF Indica si esta listo para empieza a transmitir en DMA.
- Bit 4 SERVICE / DSC, este bit indica si necesita o no un servicio o una interrupción
- Bit 3 DRQ Data Request, indique que el comando necesita escribir/leer datos.

- Bit 2 CORR, Indica si ha ocurrido algún error
- Bit 0 CHECK, Indica que ha ocurrido un error en la ejecución de un comando.

Registro de Error (ERR). Es solo de lectura, pero cuando se escribe en el se convierte en el registro de features. Este registro permite dar a conocer si existe algún error en la ejecución de comandos o en el normal funcionamiento del dispositivo.

D7	D6	D5	D4	D3	D2	D1	D0
Sense Key				MCR	ABRT	EOM	ILI

Tabla 3.8 Registro error

Bits 7-4 Sense Key⁶, indica cierto error o acción encontrada.

Bit 3 MCR Media Change Requested.

Bit 2 ABRT Aborted Command.

Bit 1 EOM End Of Media Detected.

Bit 0 ILI Illegal Length Indications.

Registro de control (CTR). Este registro permite el control de dispositivos en cuanto tiene que ver a interrupciones y al software reset.

D7	D6	D5	D4	D3	D2	D1	D0
Reserved				1	SRST	nIEN	0

Figura 3.10 Registro de control

Bit 2 SRST, Este bit es el Software Reset.

Bit 1 nIEN, Este bit enables/disables la interrupción al host.

Registro de selección del dispositivo (HEAD). Este registro es de escritura y permite la selección del dispositivo, ya sea master o slave. DVR=0; master;

⁶ Sense key, conjunto de bits que se relaciona al estado del dispositivo ante la ejecución de comandos.

DVR=1; slave. Y el Bit 6 al colocar 0, se configura CHS, a cuando es 1; se configura como LBA.

D7	D6	D5	D4	D3	D2	D1	D0
1	Reserved	1	DRV	Reserved for SAMLUN			

Figura 3.11 Registro Head device

3.2.3.3 Atapi Packet Command

Un comando ATAPI es comunicado que va a ser enviado, cuando primero se envía el Command Packet (A0) al ATAPI CD-ROM. Es decir, el comando a ejecutarse es enviado después de A0 y consta de 12 bytes o 6 words. Cuando se envía un comando sin el command packet el comando es abortado.

El dispositivo ATAPI también el actual paquete de comandos del Standard SCSI, pero no su protocolo. Los comandos más utilizados son del documento SCSI-3, siempre y cuando tengan de 10-12 bytes.

El formato utilizado para el envío de un comando con command packet es:

Byte	Bit	7	6	5	4	3	2	1	0	
0	Operation Code									
1	Reserved				Reserved					
2	(MSB)	Logical Block Address (if required)								
3										
4										
5									(LSB)	
6	(MSB)	Transfer Length (if required) or Parameter List Length (if required) or Allocation Length (if required)								
7										
8										
9									(LSB)	
10	Reserved									
11	Reserved									

Figura 3.12 Formato de comandos ATAPI

Logical Block Address: La dirección por sectores en la cuales se va a ejecutar el comando, siempre y cuando lo requiera.

Transfer Length: El bloque de datos que se enviará al dispositivo, siempre y cuando el comando lo requiera.

Parameter List Length: El bloque de datos que el dispositivo envía al host.

Allocation Length: bloque que se reserva para datos que pudiese necesitar el dispositivo.

El envío de este comando A0 tiene las siguientes entradas o requisitos para que sea ejecutado, es decir tiene que configurarse los siguientes registros, antes de enviar el comando.

Register	7	6	5	4	3	2	1	0
Features	na	na	na	na	na	na	OVL	DMA
Sector Count	Tag					na		
Sector Number	na							
Byte count low (Cylinder Low)	Byte count limit (7-0)							
Byte count high (Cylinder High)	Byte count limit (15-8)							
Device/Head	obs	na	obs	DEV	na	na	na	na
Command	A0h							

Tabla 3.9 Parámetros de configuración para el command packet.

Una vez enviado el paquete de comandos A0, el mismo solicita el comando a ejecutar siendo este de diagnostico, escritura, lectura, entre otros.

3.2.3.4 Transferencia de comandos al dispositivo ATAPI

El paquete de comandos consta de 12 bytes o 6 palabras que son enviadas o escritas al dispositivo a través del registro de comandos.

Para lo cual el protocolo ATAPI ha dispuesto cierto procedimiento que es ejecutado por el dispositivo.

Diagrama de flujo del envío del packet command al dispositivo (Anexo 1)

Diagrama de flujo de comandos que requieren leer datos del dispositivo (Anexo 2)

Diagrama de flujo de comandos que requieren escribir datos al dispositivo (Anexo 3)

Diagrama de flujo de comandos que no requieren datos (Anexo 4)

Lista de comandos ATA/ATAPI (Anexo 5)

Lista de comandos ATAPI PACKET COMMAND (Anexo 6)

3.2.3.5 Escritura en un CD

Para escribir en un disco CD como un track, cerrando totalmente debemos seguir los siguientes pasos:

1. Configurar la página de modo de parámetros de escritura
 - a) El tipo de "Track".
 - b) El Mode/Control.
 - c) tipo de sección
 - d) El tamaño de bloque (2048-2448 bytes)
2. Con la ayuda de mode sense y mode select enviamos la configuración de write parametres mode.
3. Ejecutamos el comando write
4. Repetir si deseamos.

En la siguiente figura se puede observar claramente el proceso de escritura:

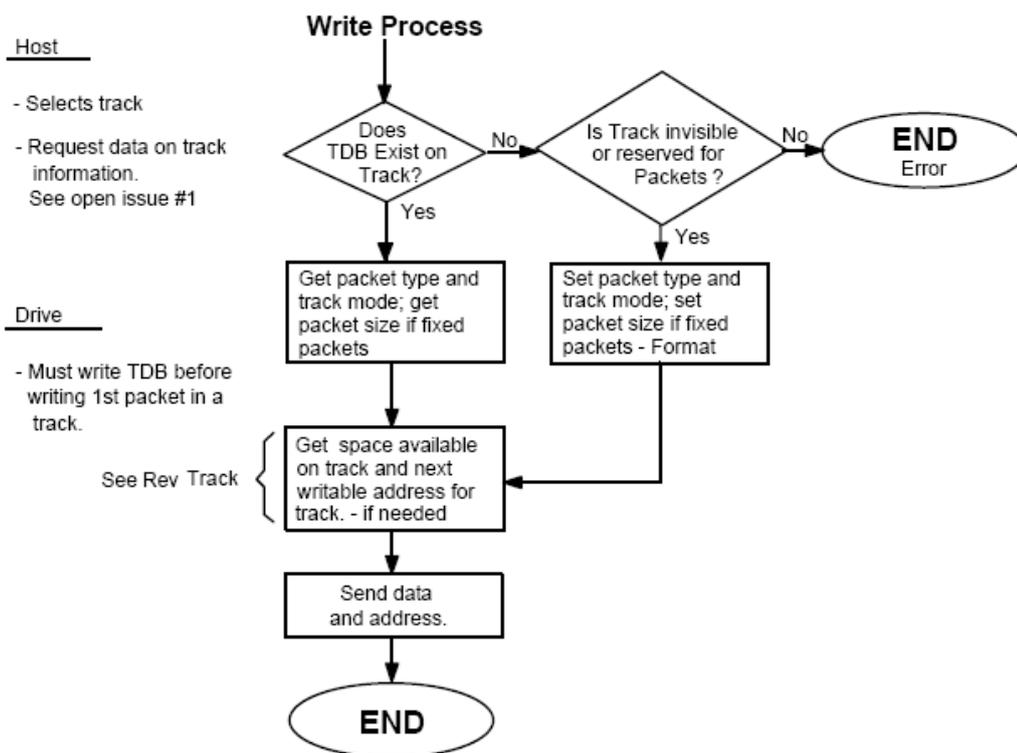


Figura 3.13 Proceso de escritura de un paquete

El comando utilizado es el Write command (10), el cuál se ejecuta con 10 bytes, como se mencionó anteriormente se debe configurar los parámetros de escritura. En la figura se encuentra el formato de este comando.

Bit	7	6	5	4	3	2	1	0
Byte								
0	OPERATION CODE (2Ah)							
1	Reserved		DPO	FUA	Reserved		RELAD	
2	(MSB) Logical Block Address (LSB)							
3								
4								
5								
6	Reserved							
7	(MSB) Transfer Length (LSB)							
8								
9	Control							

Figura 3.14 Formato del comando de escritura

DPO (Disable Page Out) bit, prioridad en la ejecución desde la pagina de caché.

FUA (force unit access) bit, cuando =1, forzar el ingreso al CD, escribe únicamente en el primer LBA. Cuando = 0, indica que el comando se ha ejecutado satisfactoriamente.

Logical Block Address (LBA), especifica la dirección de donde va empezar a escribir.

RELADR bit se encuentra en 0.

The transfer length field, especifica el numero de bloque continuos (2048*# bloques) que va ha escribir en el CD

3.2.3.6 Lectura de un CD

El READ CD Command es un comando genérico que lee todo tipo formatos de CD. Este comando puede retornar: data streams, headers, EDC, ECC, user data y CD-DA data. Y se puede configurar para cada tipo de dato a recibir. En la figura se observa el formato del comando:

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (BEh)							
1	RESERVED			Sector Type			Reserved	RELADR
2	(MSB) Starting Logical Block Address (LSB)							
3								
4								
5								
6	(MSB) Transfer Length in Blocks (LSB)							
7								
8								
9	SYNC	Header Codes	User Data	EDC & ECC	Error Field	Reserved		
10	Reserved				Sub-Channel Selection Bits			
11	Control							

Figura 3.15 Formato del comando de lectura

RELADR bit se encuentra en 0.

Sector type field es utilizado como filtro para seleccionar el tipo de formato de datos a retornar. Si (000b), retorna lee todos los formatos (CD-DA)

Sync bit, este bit incluye o no los datos de sincronización.

Header Code field Estos bits incluyen o no la información de los Header/Sub-header.

User data bit, Retorna los datos de usuario.

EDC and ECC bit, incluye los datos de control.

Error field, incluye los bloque de error que se haya generado.

3.2.4 IDE interface

Es la típica interfase para la conexión de discos duros, CD-ROM drives a una PC. En la mayoría de las motherboards se encuentra dos conectores IDE, con su propio controlador cada una de ellas. Comúnmente están definidas con las siguientes direcciones I/O:

Primary IDE controller: 1F0h a 1F7h y 3F6h

Secondary IDE controller: 170h a 177h y 376h

Cada una corresponde a un registro del controlador del IDE, y estos son utilizados por el protocolo ATA/ATAPI para el envío de instrucciones. El conector físicamente consta de 40 pines distribuidos de la siguiente manera:

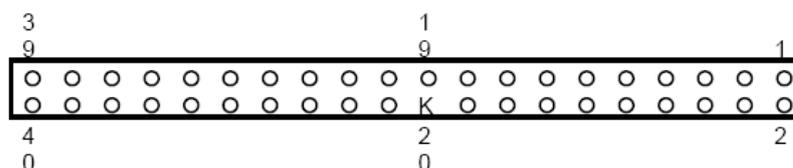


Figura 3.16 Conector IDE

Pin	Signal	Name	Source
1	RESET-	RESET	I
2	Ground	Ground	I/O
3	DD7	DATA BUS BIT 7	I/O
4	DD8	DATA BUS BIT 8	I/O
5	DD6	DATA BUS BIT 6	I/O
6	DD9	DATA BUS BIT 9	I/O
7	DD5	DATA BUS BIT 5	I/O
8	DD10	DATA BUS BIT 10	I/O
9	DD4	DATA BUS BIT 4	I/O
10	DD11	DATA BUS BIT 11	I/O
11	DD3	DATA BUS BIT 3	I/O
12	DD12	DATA BUS BIT 12	I/O
13	DD2	DATA BUS BIT 2	I/O
14	DD13	DATA BUS BIT 13	I/O
15	DD1	DATA BUS BIT 1	I/O
16	DD14	DATA BUS BIT 14	I/O
17	DD0	DATA BUS BIT 0	I/O
18	DD15	DATA BUS BIT 15	I/O
19	Ground	Ground	
20	N.C.	(Coding Pin)	
21	DMARQ	DMA REQUEST	O
22	Ground	Ground	
23	DIOW-	I/O WRITE	I
24	Ground	Ground	
25	DIOR-	I/O Read	I
26	Ground	Ground	
27	IORDY	I/O Channel Ready	
28	SPSYNC:CSEL	SPINDLE SYNC or CABLE SELECT	
29	DMACK-	DMA ACKNOWLEDGE	I
30	Ground	Ground	
31	INTRQ	INTERRUPT REQUEST	O
32	IOCS16-	16 BIT I/O	O
33	DA1	ADDRESS BIT 1	I
34	PDIAG	PASSED DIAGNOSTICS	O
35	DA0	ADDRESS BIT 0	I
36	DA2	ADDRESS BIT 2	I
37	CS0-	CHIP SELECT 0	I
38	CS1-	CHIP SELECT 1	I
39	DASP-	DRIVE ACTIVE	O
40	Ground	Ground	

Tabla 3.10 Pines del conector IDE

3.2.5 Protocolo NMEA

El protocolo NMEA es usado para la comunicación entre dispositivos de uso marino para transmitir datos. La mayor parte de los receptores envían de forma continua sentencias NMEA 0183, normalmente un grupo por segundo. Dichas sentencias, comienzan siempre por el signo “\$” y están formadas por letras, números y signos en código ASCII y termina con <CR><LF> (CR: Carriage Return, LF: Line Feed). La velocidad típica de este protocolo es de 4800 baudios.

Los primeros dos caracteres después de “\$” son los que identifican el equipo, y los siguientes tres caracteres es el identificador del tipo de sentencia que se está enviando. Los tres tipos de sentencias NMEA que existen son los de envío (**Talker Sentences**), Origen del equipo (**Proprietary Sentences**) y consulta (**Query Sentences**). Los datos están delimitados por coma, deben incluirse todas las comas, ya que actúan como marcas. Por ejemplo “GP” que se usa para identificar los datos GPS.

Los mensajes y tramas de salida se muestran en la siguiente tabla:

Mensaje	Descripción
GGA	Time, position and fix type data.
GLL	Latitude, longitude, UTC time of position fix and status.
GSA	GPS receiver operating mode, satellites used in the position solution and DOP values.
GSV	The number of GPS satellites in view satellite ID numbers, elevation, azimuth, and SNR values.
MSS	Signal-to-noise ratio, signal strength, frequency, and bit rate from a radio-beacon receiver.
RMC	Time, date, position, course and speed data.
VTG	Course and speed information relative to the ground.
ZDA	PPS timing message (synchronized to PPS).
150	OK to send message.

Tabla 3.11 Mensajes y tramas de salida del GPS

El transporte del mensaje NMEA hacia el receptor es de acuerdo al siguiente formato:

Start Sequence	Payload	Checksum	End Sequence
\$PSRF<MID>1	Data2	*CKSUM3	<CR> <LF>4

Tabla 3.12 Transporte de Mensaje

3.2.5.1 Estructura de la trama NMEA para sentencia RMC (Recommend Minimum Specific Data)

\$GPRMC,161229.487,A,3723.2475,N,12158.3416,W,0.13,309.62,120598, ,*10

Name	Example	Units	Description
Message ID	\$GPRMC		RMC protocol header
UTC Time	161229.487		hhmmss.sss
Status	A		A=data valid or V=data not valid
Latitude	3723.2475		ddmm.mmmm
N/S Indicator	N		N=north or S=south
Longitude	12158.3416		dddmm.mmmm
E/W Indicator	W		E=east or W=west
Speed Over Ground	0.13	knots	
Course Over Ground	309.62	degrees	True
Date	120598		Ddmmyy
Magnetic Variation		degrees	E=east or W=west
Mode	A		A=Autonomous, D=DGPS, E=DR
Checksum	*10		
<CR> <LF>			End of message termination

Tabla 3.13 Estructura del mensaje RMC

3.2.5.2 Habilitación del puerto serial del GPS

Este es un comando de mensaje para configurar el protocolo NMEA o SIRF en cuanto tiene que ver con los parámetros de comunicación (Baudios, bits de datos, bits de parada, y paridad) y habilitar el puerto serial del dispositivo.

Generalmente este cuando es recibido correctamente este comando, el receptor envía inmediatamente los mensajes de salida.

El ejemplo del formato de este comando es el siguiente:

`$PSRF100,0,9600,8,1,0*0C`

Name	Example	Units	Description
Message ID	\$PSRF100		PSRF100 protocol header
Protocol	0		0=SiRF binary, 1=NMEA
Baud	9600		4800, 9600, 19200, 38400
DataBits	8		8,7
StopBits	1		0,1
Parity	0		0=None, 1=Odd, 2=Even
Checksum	*0C		
<CR> <LF>			End of message termination

Tabla 3.14 Configuración del puerto serial GPS

3.2.5.3 Selección y control de velocidad de transmisión de trama

Este comando es enviado para el control de los mensajes de salida estándar NMEA GGA, GLL, GSA, GSV, RMC, y VTG. Utilizando este comando se puede configurar para recibir un solo mensaje o varios, y estos periódicamente cada segundo. Para el siguiente ejemplo tenemos:

Salida del mensaje GGA con habilitación de checksum (ejemplo en la Tabla):

`$PSRF103,00,01,00,01*25`

Habilitación del mensaje VTG con una frecuencia de 1 Hz, con checksum:

`$PSRF103,05,00,01,01*20`

Deshabilitación del mensaje VTG : \$PSRF103,05,00,00,01*21

Name	Example	Units	Description
Message ID	\$PSRF103		PSRF103 protocol header
Msg	00		See Table
Mode	01		0=SetRate, 1=Quero
Rate	00	seconds	Output—off=0, max=255
CksumEnable	01		0=Disable Checksum, 1=Enable Checksum
Checksum	*25		0=None, 1=Odd, 2=Even
<CR> <LF>			End of message termination

Tabla 3.15 Control de velocidad y mensajes del GPS.

Value	Description
0	GGA
1	GLL
2	GSA
3	GSV
4	RMC
5	VTG
6	MSS (if internal beacon is supported)
7	Not defined
8	ZDA (if 1PPS output is supported)
9	Not defined

Tabla 3.16 Mensajes de salida del GPS

3.2.5.4 Selección de Datos

La característica inicial de los receptores GPS en cuanto tienen que ver a su velocidad, fecha y hora son calculados desde los sistemas base en la tierra.

Los datos por default es el WGS 84 (World Geodetic System 1984).

Por ejemplo se tiene:

Selección de datos TOKYO_MEAN

\$PSRF106,178*32

Name	Example	Units	Description
Message ID	\$PSRF106		PSRF106 protocol header
Datum	178		21=WGS84 178=TOKYO_MEAN 179=TOKYO_JAPAN 180=TOKYO_KOREA 181=TOKYO_OKINAWA
Checksum	*32		
<CR> <LF>			End of message termination

Tabla 3.17 Selección de datos del GPS

3.3 SOFTWARE

La necesidad de elaborar una interfaz que nos permita interpretar los datos adquiridos por el sistema ha planteado la elaboración del software respectivo con el uso de un lenguaje de programación, dadas las utilidades que nos brinda Gambas por ser un lenguaje de programación con GPL GNU (General Public Licence) se ha tomado como el software de desarrollo.

3.3.1 GPL GNU (General Public Licence)

La **GNU GPL** (*General Public License* o licencia pública general) es una licencia creada por la Free Software Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

La Licencia Pública General GNU es una licencia libre, de copia permitida (copyleft) para software y otros tipos de trabajos.

Las licencias de la mayoría del software y otros trabajos prácticos están diseñadas para quitar la libertad de compartir y modificar trabajos. Por el contrario, la Licencia Pública General GNU pretende garantizar la libertad para compartir y modificar todas las versiones de un programa y asegurar que permanecerá como software libre para todos sus usuarios. La Fundación Software Libre, usa la Licencia Pública General GNU para la mayoría de software; también se aplica a cualquier otro trabajo liberado de esta forma por sus autores.

3.3.2 GAMBAS

Gambas es un lenguaje de programación libre derivado de BASIC, desarrollado por Benoit Minisini. Es similar al producto de Microsoft Visual Basic y se distribuye con licencia GNU GPL.

Permite crear formularios, botones de comandos, cuadros de texto o enlazar bases de datos como MySQL, PostgreSQL o SQLite.

Es una herramienta sencilla e interesante para estudiantes de programación visual o gráfica.

Gambas es un entorno de desarrollo que se distribuye con la licencia GPL GNU (*General Public Licence*). Esto significa que se distribuye siempre con el código fuente y respeta las cuatro libertades que propone la Fundación de Software Libre.

- La libertad de usar un programa con cualquier propósito (*Libertad 1*).
- La libertad de estudiar como funciona el programa y adaptarlo a las propias necesidades (*Libertad 2*).
- La libertad de distribuir copias con las que se pueda ayudar al vecino (*Libertad 3*).

- La libertad de mejorar el programa y hacer publicas las mejoras a los demás, de modo que toda la comunidad se beneficie (*Libertad 4*).

Las aplicaciones desarrolladas con Gambas pueden o no acogerse a la licencia GPL. Cualquier programador es libre de alterar el propio lenguaje y modificarlo a su gusto, siempre y cuando entregue el código correspondiente a esas modificaciones y respete los derechos de autor de los desarrolladores originales.

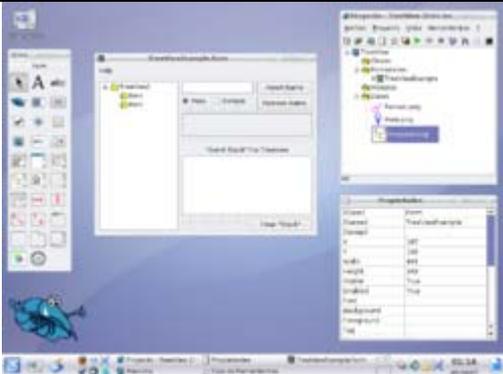
Gambas	
	
Gambas 1.0.15	
Desarrollador:	Benoit Minisini
Última versión:	1.0.18 (28 de mayo de 2007)
S.O.:	GNU/Linux
Género:	Lenguaje de programación
Licencia:	GPL
En español:	Sí
Sitio Web:	gambas.sourceforge.net

Tabla 3.18 Características de Gambas.

Gambas brinda facilidades al programador como las siguientes:

- Una ayuda muy completa del lenguaje y cada uno de los componentes.
- Una API (Interfaz para programar la aplicación) sencilla y bien documentada, lo que facilita a los programadores crear nuevos componentes para Gambas.

3.3.2.1 Desarrollo

Gambas surgió como respuesta a la necesidad de tener un entorno de desarrollo rápido de aplicaciones o RAD, además cumple la necesidad de muchos programadores del lenguaje de Microsoft de tener un lenguaje de programación sencillo para plataformas libres (como GNU/Linux y *BSD).

A pesar de estar basado en el lenguaje BASIC y poseer gran similitud con Visual Basic, Gambas no es un clon de este último.

Es importante señalar que prácticamente todos los proyectos desarrollados en este entorno son compatibles con versiones nuevas de Gambas, no así los hechos por MS Visual Basic, es decir, por ejemplo un proyecto hecho en ms Visual Basic 5.0 difícilmente correrá en la última versión, ya que no es orientado a Objetos, por lo tanto carece de herencia.

3.3.2.2 Elementos de Gambas

Para desarrollar una aplicación en Gambas son necesarios los siguientes elementos:

- Un *compilador*, que se encargará de transformar todo el código fuente y archivos que formen parte del proyecto.
- Un *intérprete* capaz de hacer que los programas hechos en Gambas sean ejecutados por el sistema operativo.

- Un *entorno de desarrollo* que facilite la programación y diseño de las interfaces gráficas de los programas.
- *Componentes* que añaden funcionalidad al lenguaje.

3.3.2.3 Portabilidad

Gambas ha sido portado a casi cualquier distribución de GNU/Linux, a excepción de Linspire en la cual hay problemas con las librerías Qt. Además, ha sido portada a OpenBSD y es mantenido como un *port oficial* de este sistema operativo.

3.3.2.4 Multiplataforma

- Funciona a la perfección en Linux x86
- Según parece también en FreeBSD
- En Windows funciona bajo KDEcygwin, es cuestión de esperar a que se libere QT4.
- En pruebas en Solaris y MacOSX (por el momento no funciona)

3.3.2.5 Cómo obtener Gambas

Las versiones de Gambas se publican a través de la pagina Web oficial del proyecto: <http://gambas.sourceforge.net>. En la actualidad existen dos versiones de Gambas la rama estable o 1.0 y la rama de desarrollo o 1.9 que desembocará en la versión 2.0.

3.3.2.6 Diferencias con Visual Basic (6.0)

La principal diferencia con respecto a Visual Basic 6.0 es el soporte para orientación a objetos. En Gambas, no existen los arrays de controles muy populares en Visual Basic pero dispone de un elemento similar llamado *group*, que utilizado conjuntamente con *arrays* de objetos ofrecen resultados análogos.

Gambas añade algunas propiedades nuevas en algunos de sus controles, como la alineación vertical/horizontal en etiquetas, o algunas propiedades gráficas en el elemento *Button* (Equivalente a *CommandButton* de Visual Basic).

El acceso a bases de datos es sumamente sencillo y potente. Gambas incluye, al igual que Visual Basic, su propio empaquetador para la distribución de aplicaciones, que permite crear incluso paquetes deb. Los componentes suministrados en Gambas 2.0 son realmente potentes, desde una librería para el manejo de documentos pdf, hasta componentes para imágenes, sockets e informes (equivalentes a Crystal reports u otras herramientas gratuitas alternativas).

Gambas es una excelente y potente alternativa a la programación visual ofrecida en Visual Basic para entornos Linux. Su deficiencia actualmente es su escasa documentación.

CAPITULO IV

DISEÑO E IMPLEMENTACIÓN DEL SISTEMA DE CONTROL DE EVENTOS EN RUTA

4.1 INTRODUCCIÓN

El presente capítulo estará centrado en el diseño e implementación de cada uno de los componentes del sistema, así como también los procedimientos, análisis y cálculos que fueron necesarios para el desarrollo del prototipo.

El primer paso, después de poseer todo el conocimiento teórico es el de obtener la información del GPS, para seguidamente separar los datos de que serán posteriormente almacenados en la memoria SD Card.

Una vez almacenada dicha información en la memoria SD Card, se adquiere la señal del sensor de velocidad y finalmente los datos que se encuentran recopilados en la SD card se graban en un CD a través del modulo CR-RW.

La información que se encuentra en el CD se la almacena en una base de datos utilizando la PC, pudiendo así determinar ubicación, velocidad, hora y fecha en la que ocurrió un evento.

El usuario podrá manipular las opciones del sistema por medio de un tablero de control que el sistema tendrá implementado.

4.2 DESCRIPCIÓN DEL SISTEMA DEL SISTEMA IMPLEMENTADO

El sistema de control de eventos en ruta para la transportación pública y de carga, obtiene y visualiza en tiempo real información de la ubicación, hora y fecha; para lo cual se utiliza un modulo GPS con los mensajes de control NMEA.

La velocidad viene a constituir una de las variables principales de este sistema, esta a su vez la obtenemos de un sensor magnético instalado en la región de la rueda del vehículo, que nos permite determinar el número de vueltas de la rueda en un segundo, y se transforma a Km./h.

Para la manipulación de estas variables se utiliza el microcontrolador **PIC16F877A**.

La información de las variables se la visualiza en un LCD (Liquid Cristal Display) de la empresa Hitachi.

Y se almacena en una memoria de seguridad **SD Card**, a través del protocolo **SPI**, y de igual manera se puede leer dicha información.

Cuando se ha completado un recorrido de viaje, el usuario tiene la opción de grabar la información en CD (Compact Disc), mediante un CD-RW 52/32/52x que utiliza el protocolo y la interfaz **ATA/ATAPI** (attachment with Packet Interface).

Un software desarrollado en plataforma Linux, con el sistema operativo **Ubuntu**, y utilizando el lenguaje de programación **Gambas**, será el encargado de leer los datos del CD y transferir a una base de Datos (**MySQL**), en donde se almacena tanto información del vehículo.

El software permite visualizar gráficamente el recorrido, puntos máximos y mínimos de velocidad, y un desplazamiento dinámico por la grafica.

4.3 DISEÑO DE HARDWARE

El controlador es el más importante en todo sistema, para el caso de este control de variables se tiene dos microcontroladores PIC16F877A, el uno que controla la adquisición de datos de las variables y el otro que gobierna el dispositivo ATAPI (CD-RW).

La descripción de pines se ha detallado en la siguiente tabla.

Función	Número de pines	Pin Número
Bus de Datos ATA	16	19-22, 27-30, 33-40
Líneas de control ATA	7	15-18, 23-24,7
Transmisión serial	2	25-26
Interfaz con PIC (1)	1	10

Tabla 4.1. Asignación de pines del PIC16F877A (2)

Función	Número de pines	Pin Número
Bus de Datos LCD	4	33-36
Líneas de control LCD	2	8,9
Transmisión serial	2	25-26
SPI	3	18,23,24
Línea de control SPI	1	17
Línea de entrada veloc.	1	6
Interfaz de usuario	6	10
Interfaz con PIC (2)	1	2

Tabla 4.2. Asignación de pines del PIC16F877A (1)

4.3.1 Generación de la señal de reloj (clock)

El circuito de reloj este compuesto por un cristal X1 de 20 Mhz, y 2 capacitores de 22 nF, que se cargan y descargan para generar la frecuencia para el funcionamiento del microcontrolador.

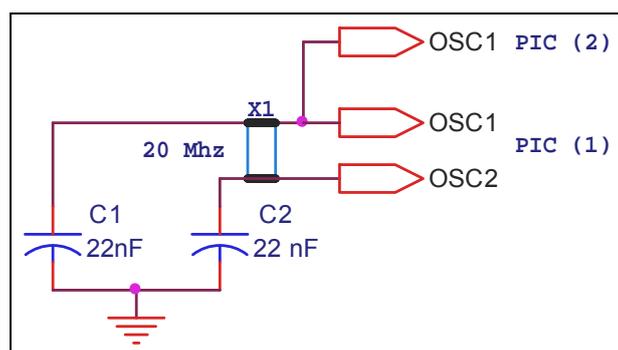


Figura 4.1. Circuito de generación de señal de reloj

Cada instrucción simple que se programa en el microcontrolador se ejecuta cada 1 us.

4.3.2 Circuito de visualización de información

EL LCD 2x16, con microcontrolador HD44780U de la marca Hitachi⁷, tiene la siguiente conexión en sus respectivos pines. Se alimenta con 5V es decir es compatible con TTL, la asignación de pines es la siguiente:

Función	Número de pines	Pin número
Bus de Datos LCD	4	11-14
Control	3	4,5, y 6
Alimentación (GND, VCC)	2	1, 2
Ajuste de contraste	1	3

Tabla 4.3. Asignación de pines del LCD

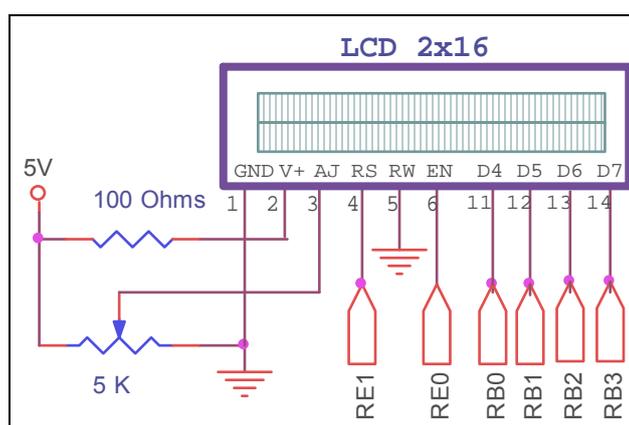


Figura 4.2. Circuito de visualización

⁷ www.hitachi.com

La resistencia de $100\ \Omega$ es para reducir la corriente que circula por el dispositivo. Se utiliza un potenciómetro o una resistencia variable de precisión de $5\ K\Omega$ para regular el contraste. RW se activa en bajo y se coloca a tierra por cuanto los comandos y datos son únicamente escritos. El bus de datos es configurado para trabajar con cuatro bits, con los MSB⁸, es decir los pines 11-14.

4.3.3 Circuito de conexión del GPS

El módulo GPS ET-102 de la empresa GlobaSat, debe tener la siguiente conexión de sus pines para el normal funcionamiento, y sus pines son asignados de la siguiente forma:

Función	Número de pines	Pin número
Canal de transmisión TXA, TXB	2	11, 14
Canal de Recepción RXA, RXB	2	15, 12
Reset	1	5
Antena	1	1
Batería backup	1	3
Alimentación (VCC)	2	2, 4
GND	4	10, 13, 16, 18

Tabla 4.4. Asignación de pines del módulo GPS

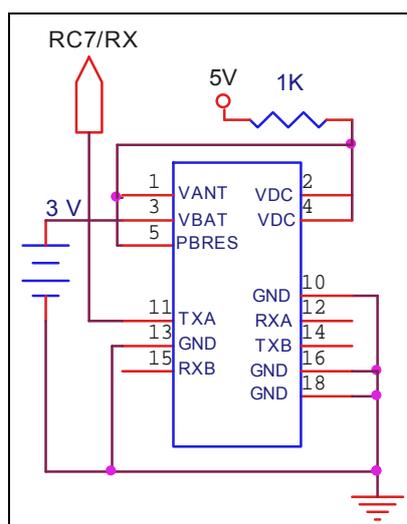


Figura 4.3. Circuito de conexión del módulo GPS

⁸ MSB, Must Significant Bits

La resistencia de $1K\Omega$ nos permite regular el voltaje de entrada VCC hacia el módulo GPS. Los pines de alimentación VDC se encuentran como un solo punto, al igual que los pines de GND.

El pin 11 que es TXA, transmite serialmente la trama GPS y esta conectada al pin 26 del microcontrolador que recibe estos datos por el puerto USART.

4.3.4 Circuito de conexión de la tarjeta SD Card

El presente circuito permite obtener el voltaje necesario, para el funcionamiento normal de una memoria SD Card, ya que esta trabaja a 3.3 V, y posee 7 pines que se les asigna de la siguiente forma:

Función	Número de pines	Pin número
Data out	1	7
Data in	1	2
Reloj (CLK)	1	5
CS (selección)	1	1
Alimentación V+	1	4
GND	2	3, 6

Tabla 4.5. Asignación de pines de la memoria SD Card

El cálculo del voltaje de 3.3 V para la memoria SD, es el siguiente:

$$R1 = 3,3k\Omega$$

$$V_{cc} = 5V$$

$$V_{SD} = 3.3V$$

$$V_o = \frac{R1 + R2}{R1 * R2} * V_{in}$$

$$5V = \frac{3,3K\Omega * R2}{3,3K\Omega + R2} * 3.3V$$

$$5K\Omega + 1.5R2 = 3,3K\Omega * R2$$

$$R2 = 1.8K\Omega$$

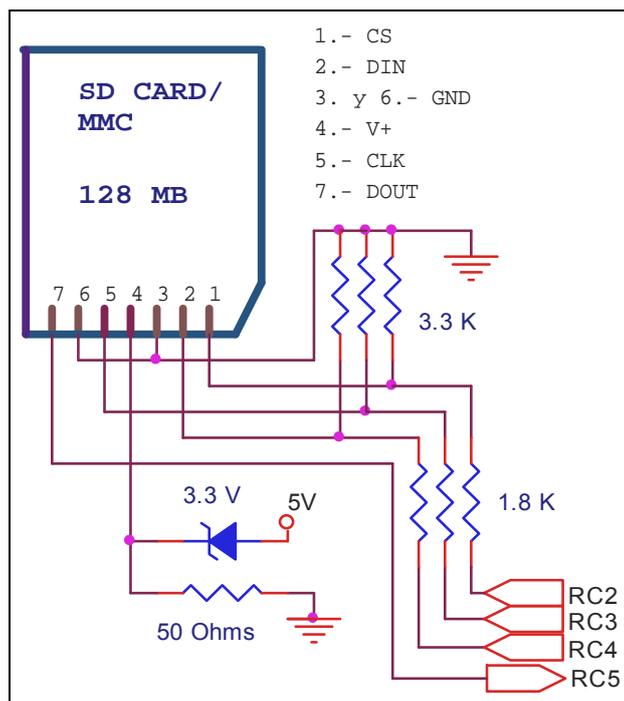


Figura 4.4. Circuito de conexión de la memoria SD Card

Las resistencias de $3,3K\Omega$ y $1,8K\Omega$ son utilizadas para generar un divisor de voltaje de 3.3 V cuando el microcontrolador envía una señal de 5V a las entradas de la memoria SD.

Estas entradas son los pines de: ingreso de datos, selección y reloj. Para el ingreso de información desde la tarjeta al PIC, se la realiza de manera directa, ya que la amplitud de la señal es de 3,3V, que el microcontrolador lo considera como un "1 lógico".

El diodo zener limita el voltaje de 5V a un máximo de 3,3 V, esta amplitud es la necesaria para la alimentación de la tarjeta. Y la resistencia de 50Ω , permite cerrar el circuito con el diodo para producir una circulación de corriente y voltaje.

4.3.5 Circuito de conexión para el dispositivo ATAPI/IDE

El conector IDE posee 40 pines los cuales permite la interfaz con el dispositivo ATAPI. En la siguiente tabla se muestra la asignación de los pines utilizados para este proyecto.

Función	Número de pines	Pin número
Bus de Datos	16	3-18
Bus de Control	7	23, 25, 33, 35, 36, 37, 38
Reset	1	1
ALE	1	28
Activación (ACT)	1	29
GND	7	2, 19, 22, 24, 26, 30, 30

Tabla 4.6. Asignación de pines del conector IDE

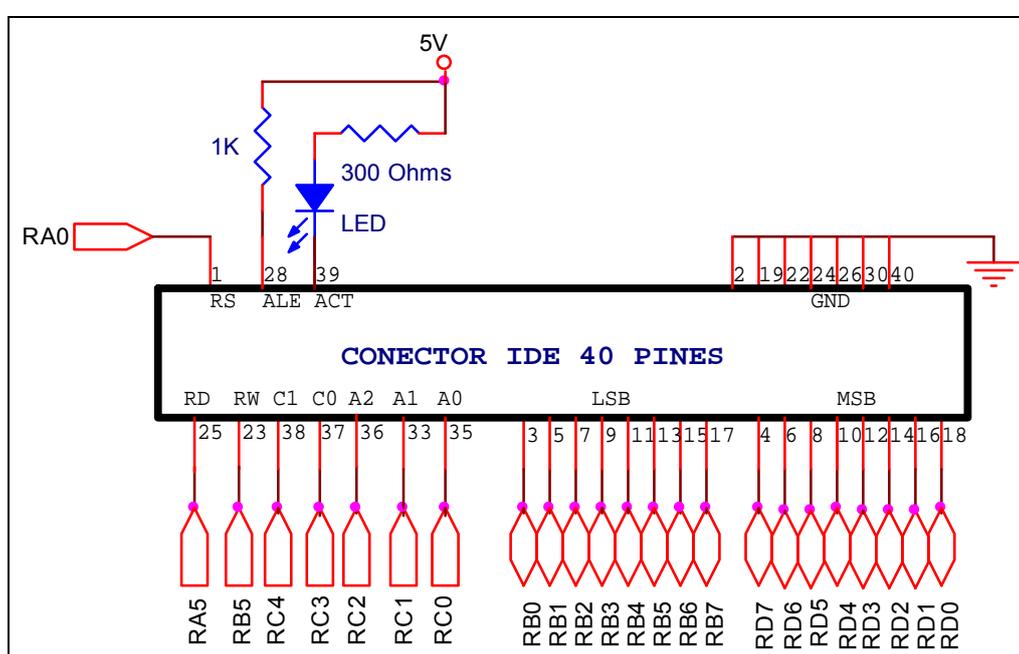


Figura 4.5. Circuito de conexión para el dispositivo ATAPI/IDE

Los pines del bus de datos, así como los de control y el pin de reset, son gobernados por el microcontrolador para cada una de las operaciones que se realizan.

Los pines de GND son conectados a la tierra común del circuito. El pin ACT indica que el dispositivo esta ejecutando alguna operación, y que el bus atapi esta siendo utilizado.

4.3.6 Circuito de comunicación Serial

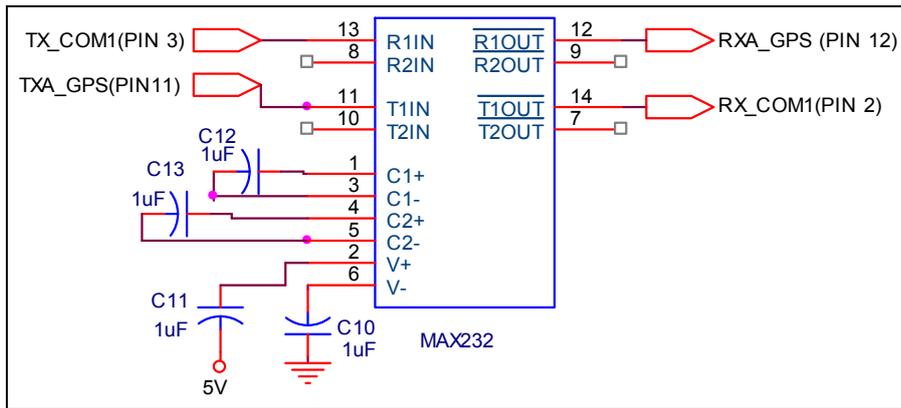


Figura 4.6. Circuito de comunicación serial.

4.3.7 Circuito de conexión del sensor de velocidad

La señal de velocidad es generada a través de un sensor, que al estar en línea con un imán se activa. Es decir únicamente se toma un pulso en cada vuelta del neumático.

En el controlador se recibe el pulso de la vuelta y se relaciona con el tiempo que se tardó en dar esta vuelta, y con la ayuda del perímetro del neumático tenemos la siguiente transformación para obtener la velocidad en km. /h.

$$e = v * t$$

$$\text{perímetro del neumático} = \text{velocidad del vehículo} * \text{tiempo de una vuelta}$$

Para el prototipo el perímetro del neumático es de 220 cm, y por ejemplo el tiempo de vuelta de de 50 ms, tendríamos los siguientes cálculos:

$$\text{velocidad} = \frac{220\text{cm}}{100\text{ms}} = 2,2\text{cm} / \text{ms}$$

$$\text{velocidad}(\text{km} / \text{h}) = \left| \frac{2,2\text{cm}}{\text{ms}} \right| * \left| \frac{1\text{km}}{100.000\text{cm}} \right| * \left| \frac{3'600.000}{1\text{h}} \right| = 79,2 \frac{\text{km}}{\text{h}}$$

$$\text{velocidad}(\text{km} / \text{h}) \approx 79 \frac{\text{km}}{\text{h}}$$

A continuación se muestra el diseño del circuito:

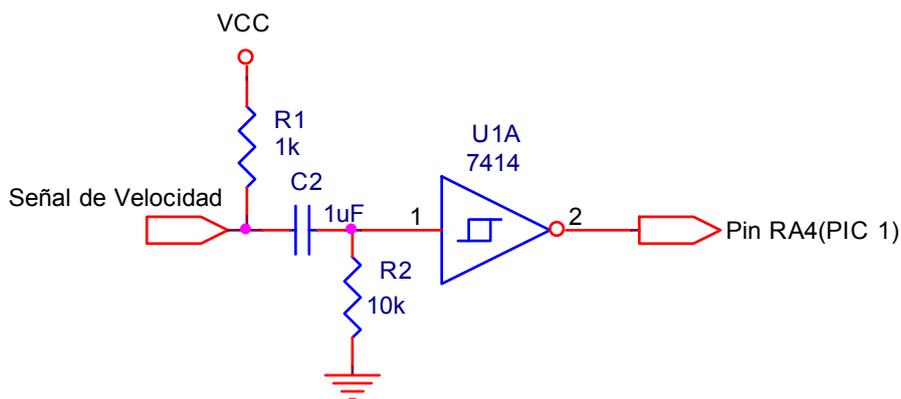


Figura 4.7 Circuito de la señal del sensor de velocidad

La principal característica de esta configuración es de ser un circuito antirebote. Cada uno de los pulsos ingresa al contador ascendente del microcontrolador a través del pin RA4.

4.3.8 Circuito de la fuente de Alimentación

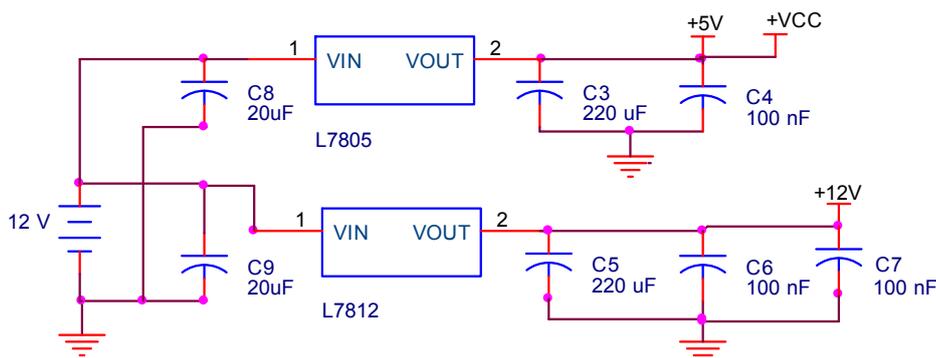


Figura 4.8 Circuito de la fuente de alimentación.

Disipación de potencia:

Cuando un regulador esta funcionando se calienta, debido a que parte de la potencia tomada del rectificador es disipada en el regulador.

La potencia disipada depende de la corriente que se esté entregando a la carga y de la caída de tensión que haya en el regulador.

Para que un regulador funcione correctamente la tensión de entrada V_{in} tenía que ser mayor que la tensión de salida V_{out} . Por lo tanto la caída de tensión en el regulador V_r será:

$$V_r = V_{in} - V_{out}$$

Y la potencia disipada vendrá dada por la siguiente ecuación:

$$PD = V_r * I_L$$

Se tiene una fuente fija con una salida de 5V y 2A, vamos a calcular la potencia que se disipa en el regulador.

Como la salida es de 5v la potencia disipada PD será:

$$PD = (12.7 - 5) * 0,5 = 3,85 \text{ W}$$

4.3.9 Interfaz de Usuario

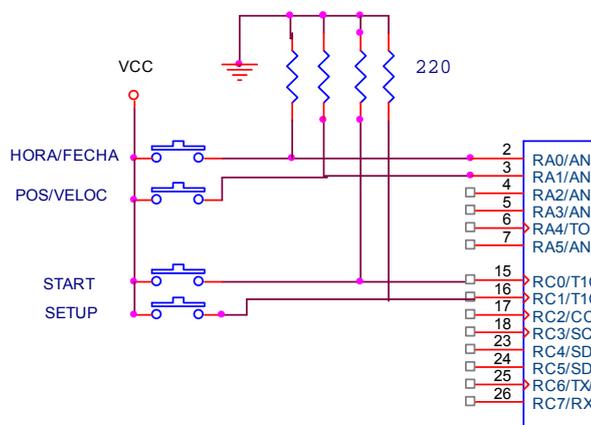


Figura 4.9 Circuito de Interfaz de Usuario

Esta compuesto por 5 botones pulsadores, un interruptor y un selector. Además en la parte visual cuenta con un LCD.

Todo empieza al encender el interruptor **POWER**, el cuál alimenta todo el sistema e inmediatamente lo inicializa teniendo el siguiente mensaje:

CONTROL

Start Setup

Al seleccionar **Start** permite iniciar la ejecución del prototipo en sus operaciones de almacenamiento y obtención de datos provenientes de las variables sensadas.

En cambio al presionar **Setup**, ingresa a la opción de configurar el sistema que únicamente esta implementado la de ingresar el número de placa del vehículo.

Al finalizar la opción configurar el sistema regrese al mensaje inicial.

Para grabar la información en un CD, se debe presionar nuevamente **Start**, visualizando el mensaje “**Desea Grabar CD? Setup: YES; Start: NO**”, para lo cual si presionamos **Setup** se procede a grabar la información y en caso de presionar **Start** el sistema continua con la adquisición de datos y su respectivo almacenamiento.

Al finalizar la operación de escritura en CD, el sistema regresa al mensaje inicial, para un nuevo recorrido.

Para visualizar la información se dispone de dos pulsadores **HORA/FECHA** y **POS/VELOC.**, los cuales al ser presionados nos coloca en el LCD la información respectiva de cada uno de estos pulsadores.

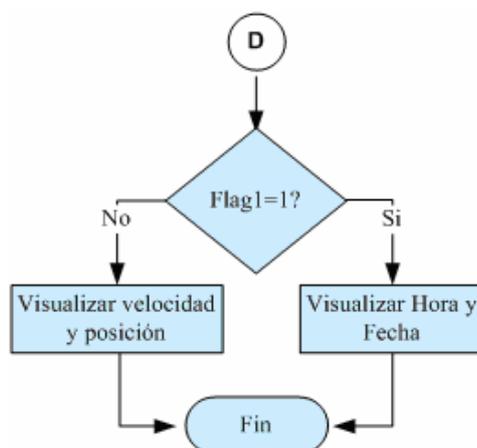


Figura 4.10 Visualización de Datos

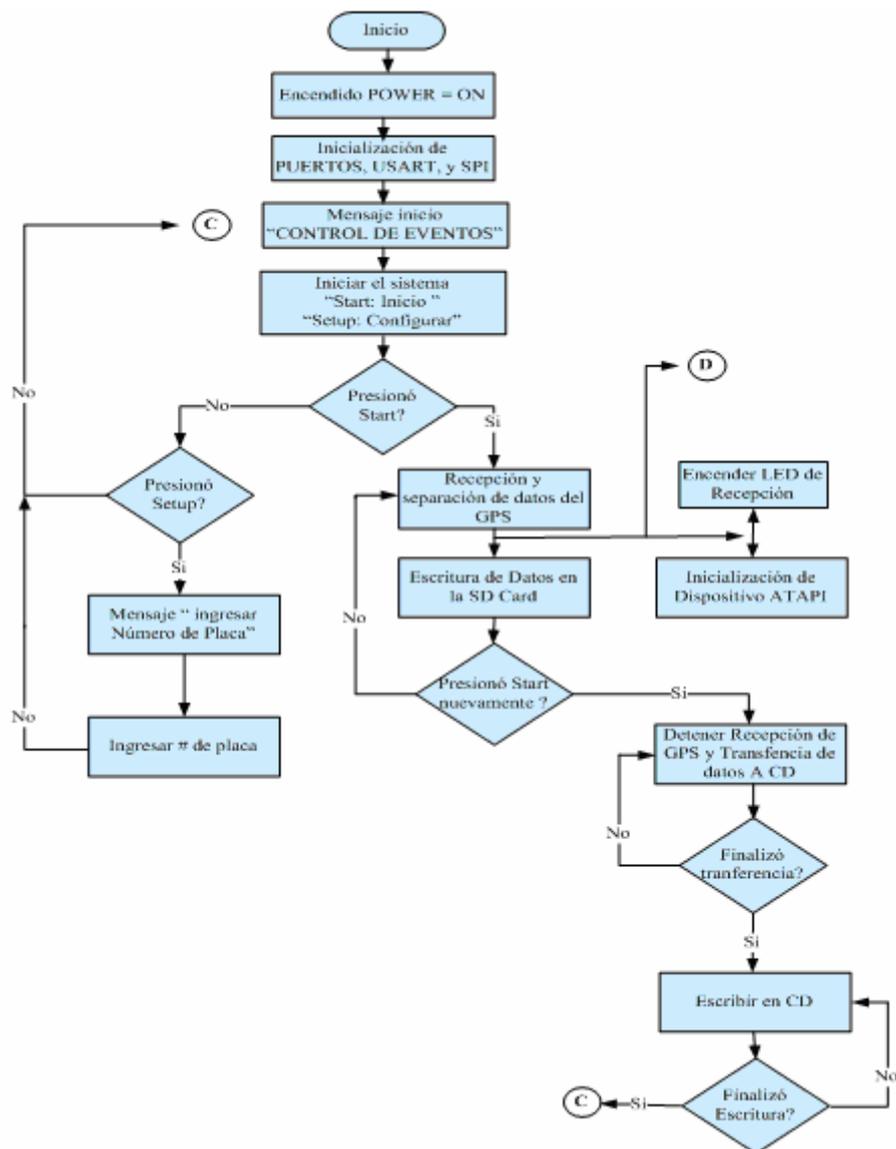


Figura 4.11 Interfaz de Usuario

4.4 PROGRAMACIÓN DEL MICROCONTROLADOR (PIC16F877A)

4.4.1 Inicialización de puertos comunicaciones y puertos físicos

Esta función permite configurar los puertos físicos de acuerdo a nuestras necesidades.

```
void SystemInit(){
```

```

    INTCON= 0; // Interrupts off
    OPTION_REG.7=1; // Deshabilita el pullpul del puerto B
    ADCON0 = 0b.0000.0001; // Configuración del registro de control del ADC

```

```

    ADCON1= 0b.0000.0110;           // Configuración de las entradas y salidas digitales
    TRISA = 0b.0011.1110;           // Configuración analógica y digital del puerto A
    TRISC = 0b.1001.0000;           // sck rc3-0, sdo rc5-0, CS rc2-0. TX y RX
    TRISB = 0b.0000.0000;           // Puerto B, configuración de puerto de salida
    TRISD = 0b.0000.0000;           // Puerto D, configuración de puerto de salida
    TRISE = 0b.0000.0000;           // Puerto E, configuración como salida
    lcd_init();                       // Inicialización del Display
    InitUSART();                       // Inicialización de la usart
    InitMMC();                          // Inicialización de la SD CARD
}

```

4.4.2 Programa de visualización del LCD

Para la visualización en el LCD, contamos con las siguientes funciones y las cuales se utiliza para este sistema de control.

Funciones: lcd_init(), lcd_write(), lcd_clear(), lcd_goto(), lcd_puts(), lcd_putchar().

4.4.2.1 Función de Inicialización del LCD

```

#pragma bit LCD_RS @PORTE.1
#pragma bit LCD_EN @PORTE.0

void lcd_init() {

    PORTE=0x00;

    DelayMs(15);                       // wait 15mSec after power applied,
    PORTB=0x03;
    LCD_EN = 1; LCD_EN = 0;
    DelayMs(5);
    LCD_EN = 1; LCD_EN = 0;
    DelayUs(200);
    LCD_EN = 1; LCD_EN = 0;
    DelayUs(200);
    PORTB = 2;                          // Four bit mode
    LCD_EN = 1; LCD_EN = 0;
    lcd_write(0x28);                     // Set interface length
    lcd_write(0xF);                      // Display On, Cursor On, Cursor Blink
    lcd_clear();                          // Clear screen
    lcd_write(0x6);                       // Set entry Mode
}

```

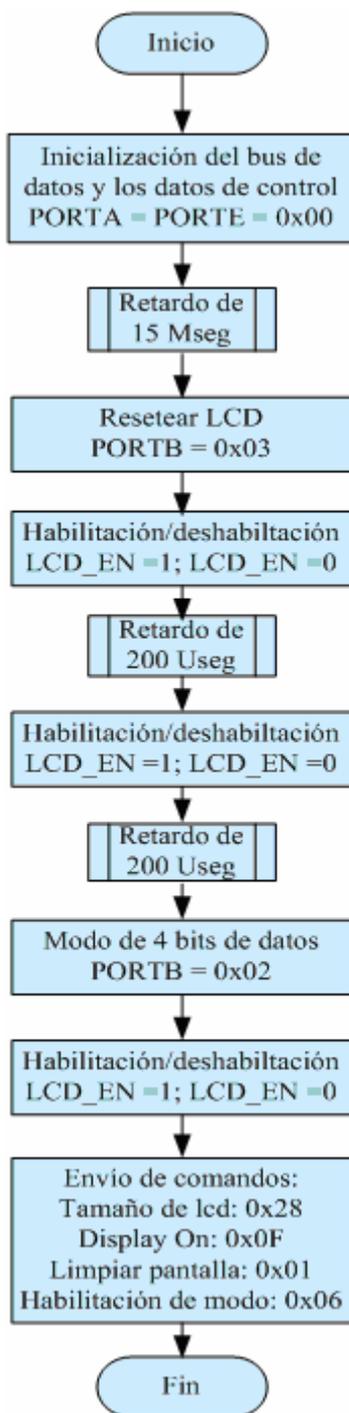


Figura 4.12. Flujograma de la inicialización del LCD

4.4.2.2 Funciones de escritura al LCD

Estas funciones permiten enviar al LCD tanto comandos como caracteres con las diferentes funciones.

/ function write char character */*

```
void lcd_write(unsigned char c)
{
    DelayMs(4);
    PORTB = ( ( c >> 4 ) & 0x0F );
    LCD_EN = 1; LCD_EN = 0;
    PORTB = ( c & 0x0F );
    LCD_EN = 1; LCD_EN = 0;
}
```

/ write a string of chars to the LCD */*

```
void lcd_puts(const char * s)
{
    LCD_RS = 1; // write string characters
    while(*s) lcd_write(*s++);
}
```

/ write one character to the LCD */*

```
void lcd_putch(char c)
{
    LCD_RS = 1; // write characters
    lcd_write( c );
}
```

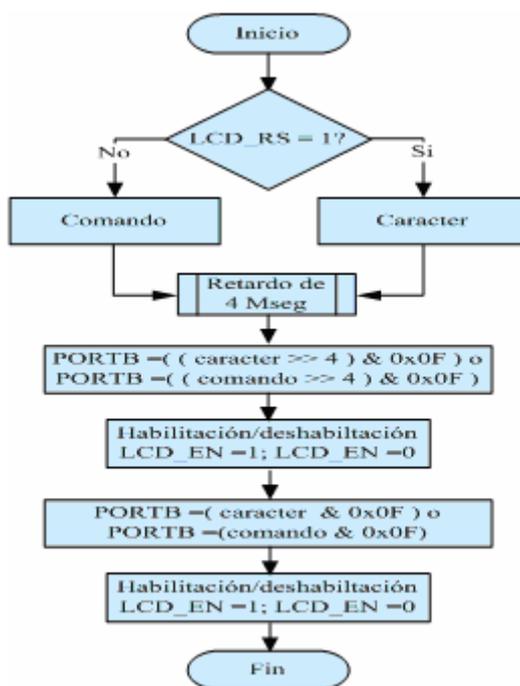


Figura 4.13. Flujograma del proceso de escritura de comando/carácter

4.4.2.3 Funciones goto y clear del LCD

/ Clear and home the LCD */*

```
void lcd_clear(void)
{
```

```

LCD_RS = 0;
lcd_write(0x1);
DelayMs(2);
}

/*Go to the specified position */

void lcd_goto(unsigned char pos)
{
    LCD_RS = 0;
    lcd_write(0x80+pos);
}

```

4.4.3 Función de retardo (Delay)

Esta función permite generar un retardo en milisegundos, el contador es un dato usn16 que permite 65535 milisegundos.

```

void DelayMs(usn16 cnt){

#if XTAL_FREQ > 2
unsigned char i;
    do {
        i = 4;
        do {
            DelayUs(250);
        } while(--i);
    } while(--cnt);
#endif
}

```

4.4.4 Función de Inicialización del USART del PIC16F877A

Esta función nos permite habilitar el puerto USART y dar características de comunicación.

```

void InitUSART()
{
    BRGH = 1; // Selección del rango de baudios, 1 = Alta velocidad
    SPBRG = 255; // (4800 baud/20MHz clock) velocidad de la transmisión
    SPEN = 1; // Habilitación de los pines del puerto serial RC6 y RC7
    SYNC = 0; // Configuración en modo asincrónico
    TX9 = 0; // Configuración de 8 bits para la transmisión
    RX9 = 0; // Configuración de 8 bits para recepción
    CREN = 1; // Habilitación para recepción
    TXEN = 1; // Habilitación para transmitir
    RCIE = 0; // Interrupción serial inhabilitada
}

```

4.4.5 Funciones de transmisión y recepción serial por el puerto USART

La función `Recieve_Data()`, recibe la trama del GPS y también recibe los datos provenientes de la SD Card cuando se selecciona grabar CD. La transmisión de da cargando los datos en TXREG.

*/*Recepción de Datos */*

unsigned char *Recieve_Data()*

{

while(!RCIF) *// RCIF: Flag de recepción por el USART*
continue; *// 1= El buffer de datos está lleno, 0 = no está lleno*
return RCREG; *// retorna el valor leído*

}

*/*Transmisión de Datos */*

while(!TXIF); *// TXIF: Flag de transmisión el USAR*
TXREG =datos;*// Send data byte, Se carga el valor y se Tx*

4.4.6 Inicialización de la SD Card o MMC

bit *InitMMC()* {

char *i;*

SMP=0; *// Fase de datos de entrada SPI; 0 = ½ tiempo*
CKE=0; *// Selección del flanco de reloj SPI, 0= asc.*
CKP=1; *// High condition passive*
SSPM0=SSPM1 = 0; *// Bit de selección del modo del SSP*
SSPEN=1; *// Bit de habilitación del módulo SSP*
CS=1; *// SD -Disabled*

for(i=0; i < 10; i++)**SPI**(0xFF); *//10*8=80 ciclos de reloj para inicialización*

CS=0; *// SD -Enabled*
if (**Command**(0x40,0,0,0x95) !=1)**goto** Fehler; *// CMD0 Reset*
st:

if (**Command**(0x41,0,0,0xFF) !=0) **goto** st; *// CMD1 incializar la tarjeta*
return 1;

Fehler:
return 0; *// Return valor verdadero*

}

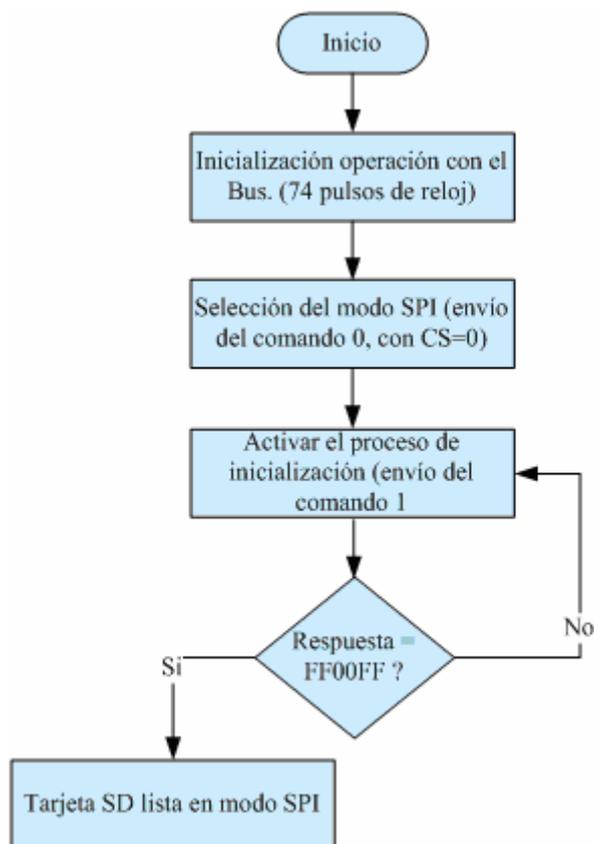


Figura 4.14. Flujograma del proceso inicialización de la SD Card

4.4.7 Transmisión de Datos a la SD Card

```

char SPI(char out) {
    SSPBUF=out;           // Cargamos el dato al buffer del SPI
    while (!BF);         // Bit estado de Buffer lleno
    return SSPBUF;       // 1= Rx completa, SSPBUF está lleno, 0= Rx no completa,
                        // SSPBUF está vacío
}
    
```

4.4.8 Envío de comandos a la SD Card

```

char Command(char befF,uns16 AdrH,uns16 AdrL,char befH){
    SPI(0xFF);           // limpiar el buffer
    SPI(befF);           // primer byte
    SPI(AdrH.high8);     // Tomamos los 8 bits mas significativos de los 16
    SPI(AdrH.low8);      // Tomamos los 8 bits menos significativos de los 16
    SPI(AdrL.high8);     // Tomamos los 8 bits mas significativos de los 16
    SPI(AdrL.low8);      // Tomamos los 8 bits menos significativos de los 16
    SPI(befH);           // Byte final
    SPI(0xFF);           // Response as a return value
    return SPI(0xFF);
}
    
```

4.4.9 Lectura de un byte de la SD Card y transmitir

Esta función se ha modificado con el fin de leer los datos de la SD Card y transmitir a través del puerto USART. Se utiliza para leer y transferir los datos hacia la etapa de grabación de información en el CD. La variable K contiene la cantidad de direcciones en las que se ha grabado.

```

void read_mmc(){
    Command(0x51,z,512,0xFF); // send command
    while(SPI(0xFF) != 0xFE); // Si es = 0xFE, comando correcto
    for(l=0; l < 512; l++){
        datos=SPI(0xFF); // Receive data
        while(!TXIF); // TXIF: Flag de transmisión el USAR
        TXREG =datos; // Send data byte, Se carga el valor y se Tx
    }
    SPI(0xFF); // 2 Bytes de verificación
    SPI(0xFF);
    z++; // se incrementa la dirección a leer
    k--; // Decremento de direcciones escritas
}
    
```

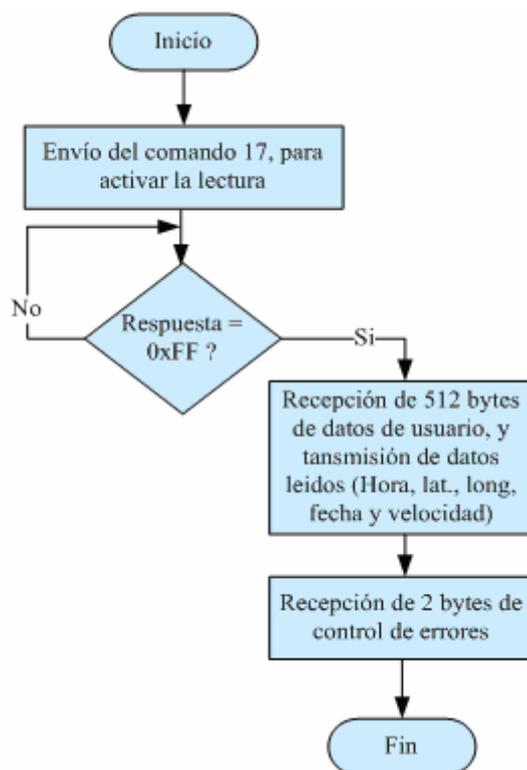


Figura 4.15. Lectura de la memoria SD Card en modo SPI

4.4.10 Escritura en una dirección de la memoria SD card

Este comando se ha modificado para escribir parte de la trama de los datos del GPS, y el valor de k es el dirección en la que se va a grabar y se incrementa hasta 65535.

```

void write_mmc(){
    Command(0x58,k,512,0xFF);           // CMD18 send command
    SPI(0xFF);                           // Datos de inicio de grabación
    SPI(0xFF);
    SPI(0xFE);
    for(j=0;j<512;j++) SPI(D_GPS[j]);    //Graba la cadena de datos del GPS
    SPI(255);                             // Se envía dos bytes de comprobación
    SPI(255);
    i=SPI(0xFF);
    while(SPI(0xFF) !=0xFF);             // mientras el bus este ocupado.
}
    
```

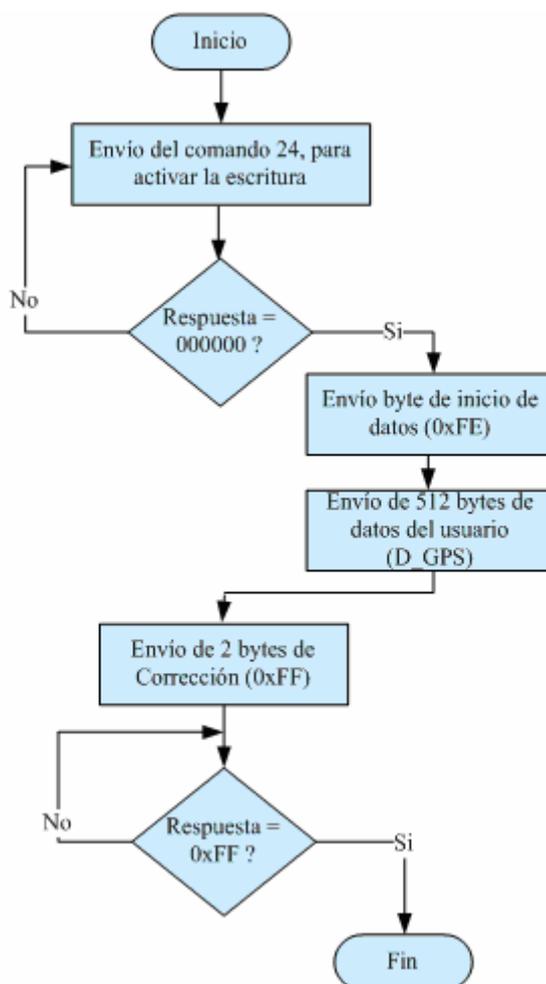


Figura 4.16. Escritura en la memoria SD Card en modo SPI

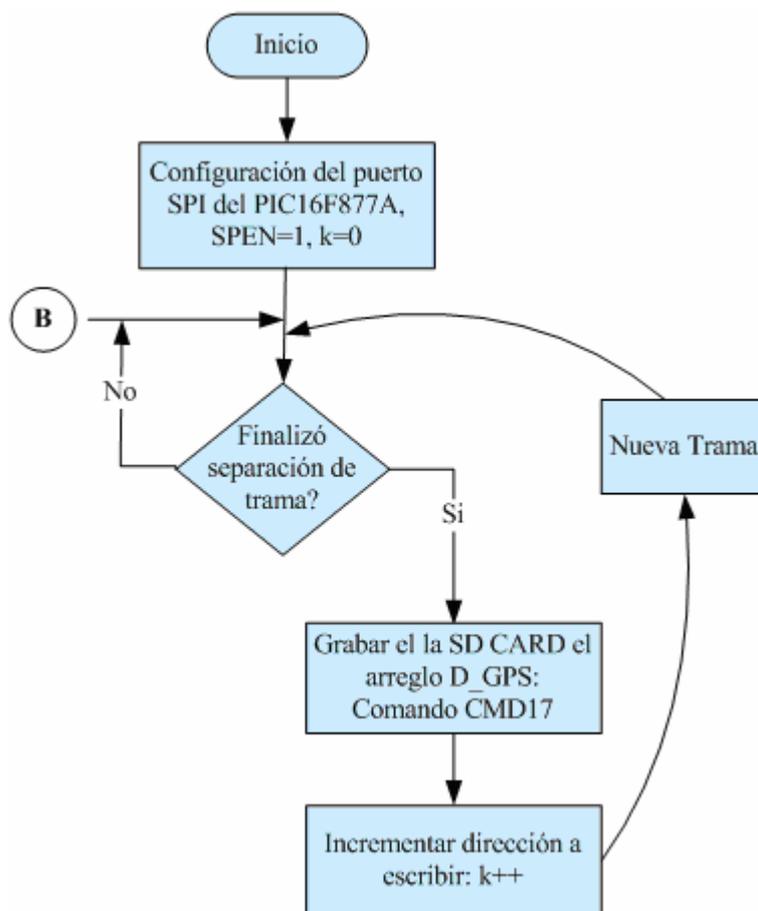


Figura 4.17. Escritura en la memoria SD Card en modo SPI

4.4.11 Inicialización y detección de un dispositivo ATAPI

```

set_register(0x00,DHE_REG_ADD); // envío de 0x00 al registro Head
set_register(0x55,SCO_REG_ADD); // envío de 0x55 al registro sector couter
set_register(0xAA,SNU_REG_ADD); // envío de 0xAA al registro sector number
set_register(0xAA,SCO_REG_ADD); // envío de 0xAA al registro sector couter
set_register(0x55,SNU_REG_ADD); // envío de 0x55 al registro sector number
set_register(0x55,SCO_REG_ADD); // envío de 0x55 al registro sector couter
set_register(0xAA,SNU_REG_ADD); // envío de 0xAA al registro sector number
    
```

```

bsy_drq(); // BSY=0; DRDY=1
    
```

```

if(get_register(SCO_REG_ADD)==0x55&&get_register(SNU_REG_ADD) == 0xaa);
    
```

```

set_register(0x00,DHE_REG_ADD);
    
```

```

if(get_register(CLO_REG_ADD) != 0x14 && get_register(CHI_REG_ADD) != 0xeb) goto rs; //
rs: salta a reset del cdrw si no le reconoce
    
```

4.4.12 Envío del packet command (A0) al dispositivo ATAPI

Antes de llamar a la función `send_packet()`, se debe configurar el arreglo `packet` en donde se encuentra los comandos y sus respectivos parámetros para su ejecución.

```
void send_packet(){
    st = get_register(STA_REG_ADD);           // STATUS
    j = 0;                                   // initiation var of data lsb
    a = 1;                                   // initiation var of data msb
    while(st.3 == 1){                         // while DRDY == 1
        a1 = packet[a];
        set_data(packet[j],a1,DTA_REG_ADD);  // send data
        st = get_register(STA_REG_ADD);      // STATUS
        j = j+2;                              // increment j
        a = a+2;                              // increment a
    }
}
```

4.4.13 Escritura de datos en un CD

Este comando permite grabar datos obtenidos de la memoria SD Card a través de la recepción desde el puerto Usart.

```
for(i = 0; i < 55; i++) packet[i] = 0;      // Initiation array packet
packet[0] = 0x2A;                          // write command code
packet[7] = 0x34;                          // length user data (128 MB max)
packet[8] = 0x00;
set_register(0xA0,CMD_REG_ADD);            // send packet command
DelayMs(100);                              // delay
send_packet();                             // send write command
CREN=1;                                    // enable receive data (usart)

do{
    st = get_register(STA_REG_ADD);         // Status register
    PORTE.1 = 1;                           // initiation RX
    if(CREN == 1){
        datos.low8 = Recieve_Data();        // Rx data
        datos.high8 = Recieve_Data();      // Rx data
    }
    else {
        datos.low8 = 'p';
        datos.high8 = 'q';
    }

    if (datos.low8 == 'F' || datos.high8 == 'F') CREN = 0;
    set_data(datos.low8,datos.high8 , DTA_REG_ADD);
} while(st.3 == 1 || st.7 == 1);           // while BSY==1 o DRDY==1

lcd_puts("Ready");
```

4.4.14 Configuración e inicialización del Módulo GPS

El módulo GPS se ha configurado mediante software, es decir accedemos a través del puerto serial de una computadora personal (PC). En el siguiente flujograma se observa la secuencia de inicialización y recepción.

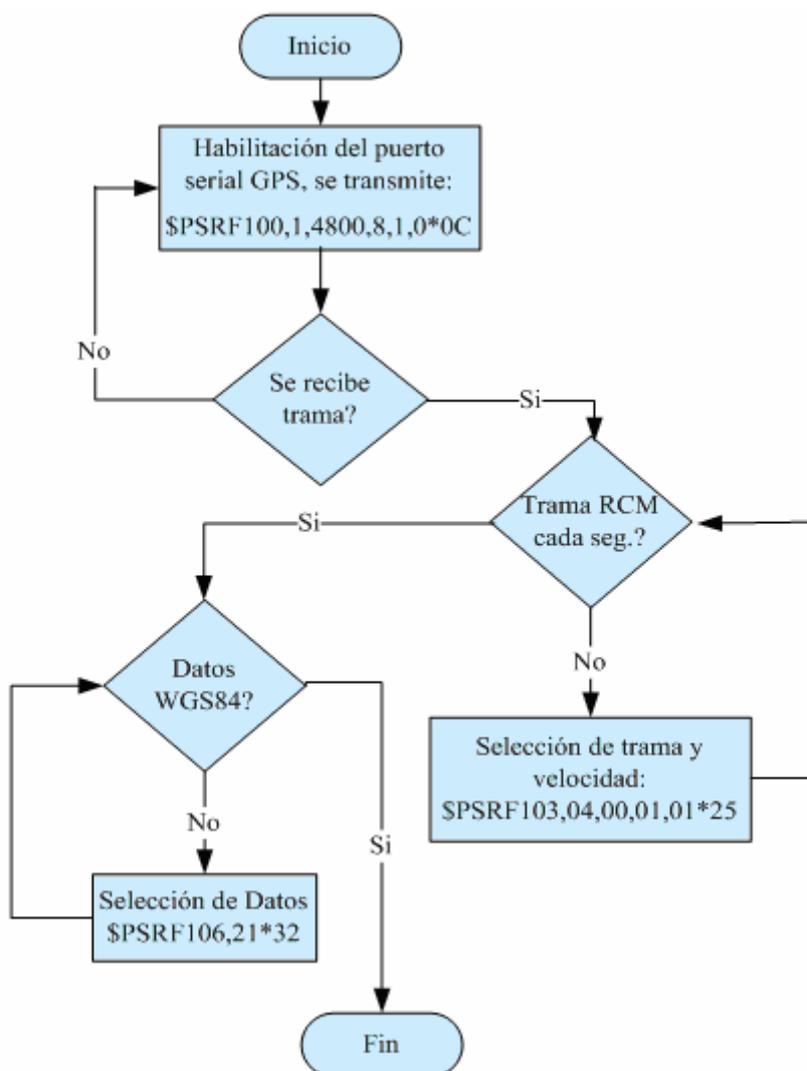


Figura 4.18. Configuración e Inicialización del módulo GPS

El software utilizado es el SIRFDemo, que permite la configuración de todos los parámetros del GPS, De esta manera es más fácil obtener rápidamente la trama. Gráficamente tenemos:

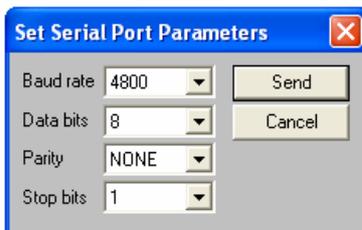


Figura 4.19. Configuración del Puerto serial GPS (Set Main Serial Port)

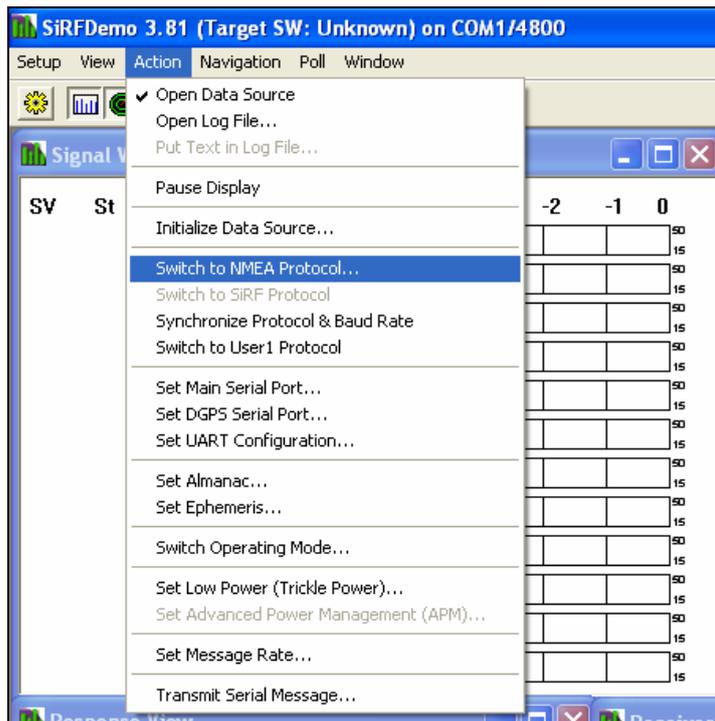


Figura 4.20. Cambio del protocolo SIRF a protocolo NMEA

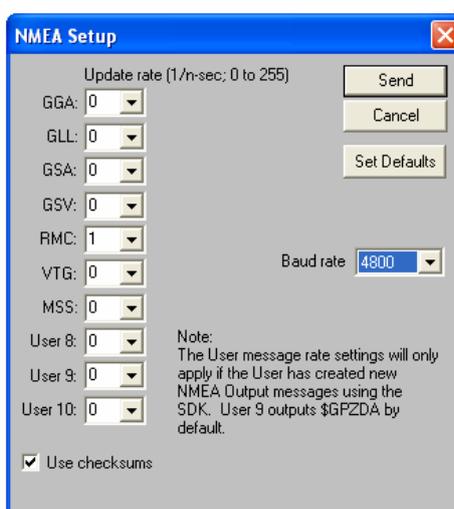


Figura 4.21. Selección de la trama a transmitir y velocidad de la misma

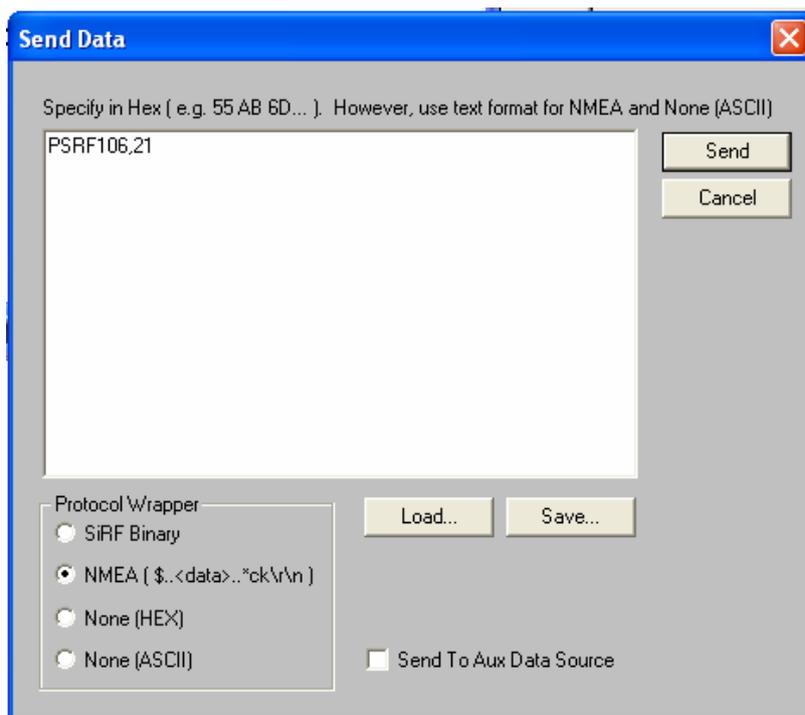


Figura 4.22. Selección del tipo de Dato a transmitir (WSG 84)

4.4.15 Configuración de USART en el PIC16F877A y Separación de trama GPS

La separación de los datos se toma como referencia la separación por comas (,) después de cada uno de los parámetros de la trama RMC.

```

if(datos=='$'){
    almacenar=1;
    f1=comas=cont=0;
}

if(datos==',' ){
    comas++;
    f2=0;
}

if(comas==10){
    D_GPS[f1]='*';
    almacenar=0;
}

if(almacenar==1){

```

```
if(comas==1){ // Tiempo: hora minutos y segundos
    if(datos==''){
        D_GPS[f1]=';';
        f1++;
        goto fin;
    }
    if(f2>5)goto fin;
    D_GPS[f1]=datos; // Se almacena los datos en D_GPS
    f2++;
    f1++;
}

if(comas==3){ // Latitud
    if(datos==''){
        D_GPS[f1]=';';
        f1++;
        goto fin;
    }
    if(f2>5)goto fin;
    D_GPS[f1]=datos; // Se almacena los datos en D_GPS
    f2++;
    f1++;
}

if(comas==4){ // Orientación cardinal de latitud
    if(datos==''){
        D_GPS[f1]=';';
        f1++;
        goto fin;
    }
    D_GPS[f1]=datos; // Se almacena los datos en D_GPS
    f1++;
}

if(comas==5){ // Longitud
    if(datos==''){
        D_GPS[f1]=';';
        f1++;
        goto fin;
    }
    if(f2>6)goto fin;
    D_GPS[f1]=datos; // Se almacena los datos en D_GPS
    f2++;
    f1++;
}

if(comas==6){ // Orientación Cardinal de longitud
    if(datos==''){
        D_GPS[f1]=';';
        f1++;
        goto fin;
    }
    D_GPS[f1]=datos; // Se almacena los datos en D_GPS
    f1++;
}

if(comas==9){ // Fecha
    if(datos==''){
        D_GPS[f1]=';';
```

```

    f1++;
    goto fin;
}
D_GPS[f1]=datos; // Se almacena los datos en D_GPS
f2++;
f1++;
}
fin:
}

```

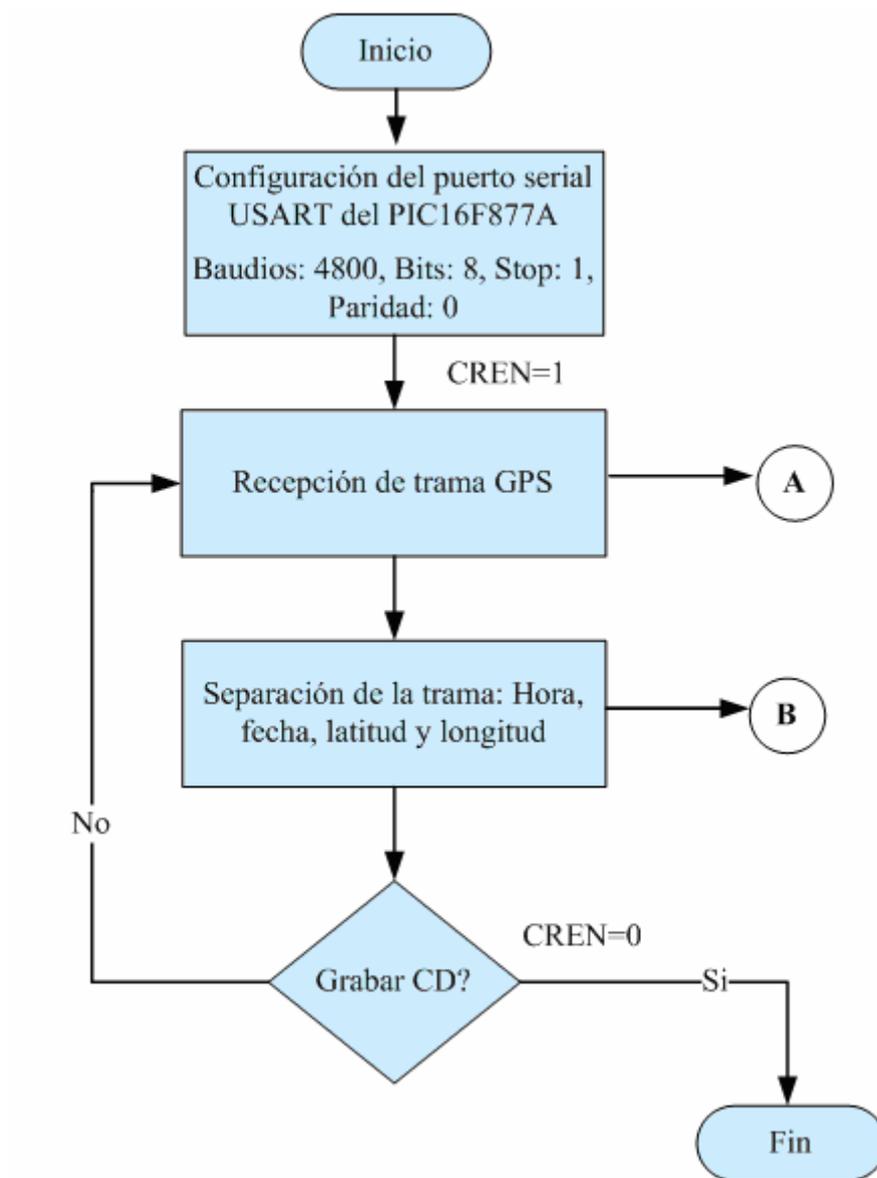


Figura 4.23. Proceso general del GPS

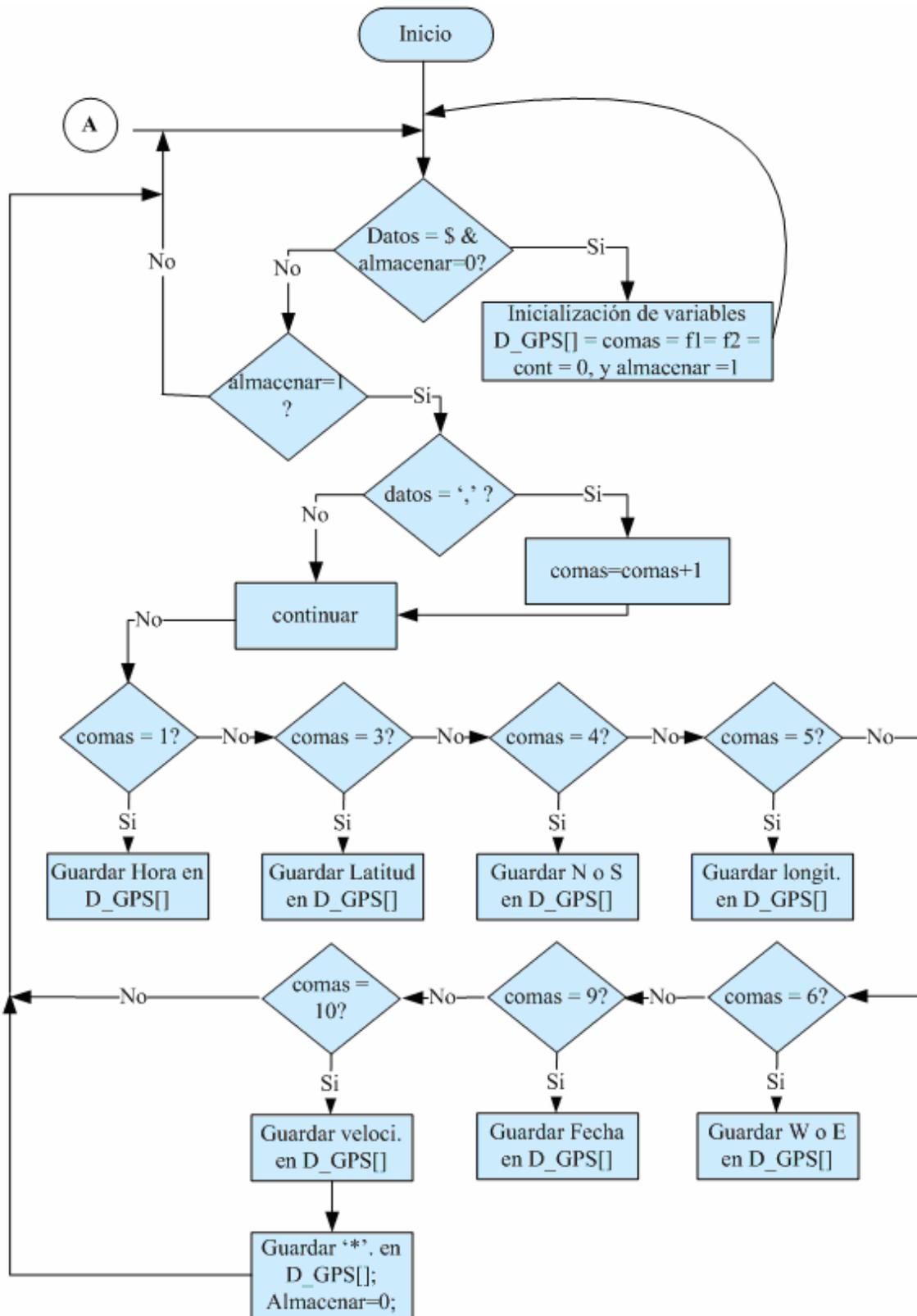


Figura 4.24. Separación trama de GPS

4.5.1.1 Pantalla de Presentación:



Figura 4.26. Pantalla de Presentación de SSV 1.0

La **Pantalla de Presentación** brinda la posibilidad de visualizar el logotipo del software realizado así como botones con las siguientes aplicaciones:

- Información.
- Acceder al siguiente formulario.
- Finalizar la aplicación.



.- Mediante la utilización del botón (info.) se desplegará una subpantalla en la cual podremos visualizar información acerca de las personas que desarrollaron el software como se muestra en la siguiente figura.

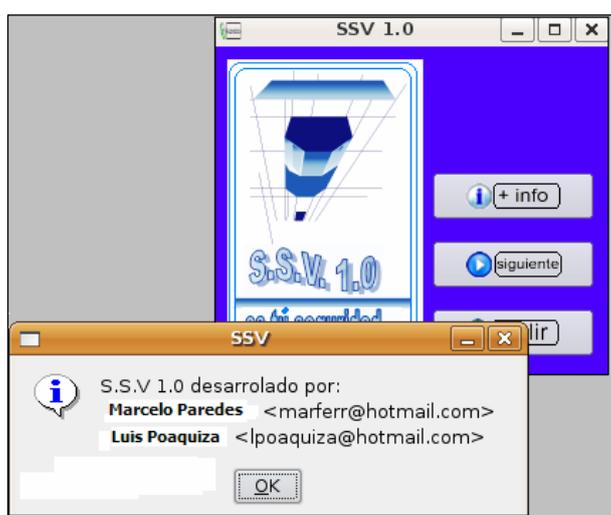


Figura 4.27. Pantalla de Información SSV 1.0



- la utilización del botón siguiente nos dará la posibilidad de cerrar la pantalla de presentación, para luego acceder a la pantalla de acceso.



- el botón salir nos da la posibilidad de cerrar la aplicación.

4.5.1.2 Pantalla de Acceso:



Figura 4.28. Pantalla de Acceso SSV 1.0

La **Pantalla de Acceso** brinda la posibilidad de visualizar el logotipo, el nombre del software, así como seleccionar los usuarios que se tengan guardados, también el ingreso de la respectiva clave.



- Esta opción brinda la posibilidad de acceder a todos los usuarios que se tengan registrados en la Base de Datos (MySQL), que mediante la conexión entre Gambas→MySQL permitirán desplegarlos en un ComboBox.



- Mediante la Caja de Texto podemos ingresar la clave de acceso que de igual manera que los usuarios estará registrada en la Base de Datos.



-El botón aceptar de la pantalla de Acceso permitirá verificar si el usuario y la clave son correctos, permitiendo registrar el nombre del usuario y la hora de acceso al sistema.



- El botón inicio de la pantalla de acceso da la posibilidad de retroceder a la Pantalla de Presentación.

4.5.1.3 Pantalla de Principal:



Figura 4.29. Pantalla Principal SSV 1.0

La **Pantalla Principal** ofrece la posibilidad de acceder a los datos del CD así como de grabar estos datos en su respectiva Base de Datos, también establece la posibilidad de acceder a datos grabados con anterioridad para ver un informe (*Gráfica de Comportamiento, datos máximos y mínimos*).



.- La Pantalla Principal presentara el nombre del usuario que se encuentra manipulando los datos ingresados.



.- El botón inicio de la pantalla principal da la posibilidad de volver a la Pantalla de Presentación.



.- el botón salir nos da la posibilidad de cerrar la aplicación.



.- la utilización del botón (info.) se desplegara una subpantalla en la cual podremos visualizar información acerca de las personas que desarrollaron el software.

La opción de la Pantalla principal en la cual se puede acceder al CD tiene las características que se muestran en la siguiente figura:



Figura 4.30. Pantalla de Ingreso de Datos de Recorrido



.- Mediante la caja de texto que se visualiza se podrá ingresar el nombre de la placa del vehiculo que hubiere realizado el recorrido.



.- Mediante la caja de texto que se visualiza se podrá ingresar el nombre del recorrido realizado por el vehiculo.



.- El botón Grabar de la Pantalla Principal permitirá acceder a los datos generados por el CD mediante una ventana de acceso, para posteriormente Grabarlos en la Base de datos que tendrá el nombre de la Placa y tablas que tendrán el nombre de cada recorrido.



Figura 4.31. Pantalla de Selección de Recorrido

Si los datos necesarios para grabar (Placa y/o Recorrido) no estarían llenos el programa mostrara un mensaje de información.



Figura 4.32. Mensaje de Advertencia (Falta de Datos)

La opción de la Pantalla principal en la cual se puede acceder a la Base de Datos tiene las características que se muestran en la siguiente figura:



Figura 4.33. Pantalla de Selección de Recorridos



- La caja de texto mostrada brindara la posibilidad de acceder a la base de ingresada que tendrá el nombre de la placa del vehiculo ingresado.



- el botón buscar dará la posibilidad de visualizar todos los recorridos realizados por el vehiculo de placa ingresada en la parte superior. Si no se hubiere ingresado la placa y se hubiese dado clic en el botón buscar aparecerá un mensaje de información que mostraría el error.

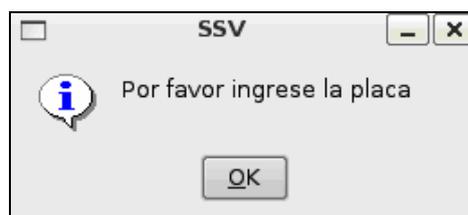


Figura 4.34. Mensaje de Advertencia (Falta de Placa)



.- una vez ingresada la placa y habiendo dado un click en el boton buscar se mostrara en un ComboBox todos los recorridos realizados por el vehiculo que tuviese la placa ingresada, pudiendo seleccionar cualquiera de los recorridos.



.- El botón conectar permitirá tener acceso a los *Datos Generales* así como a los *Informes* del vehiculo ingresado así como de un recorrido dado.

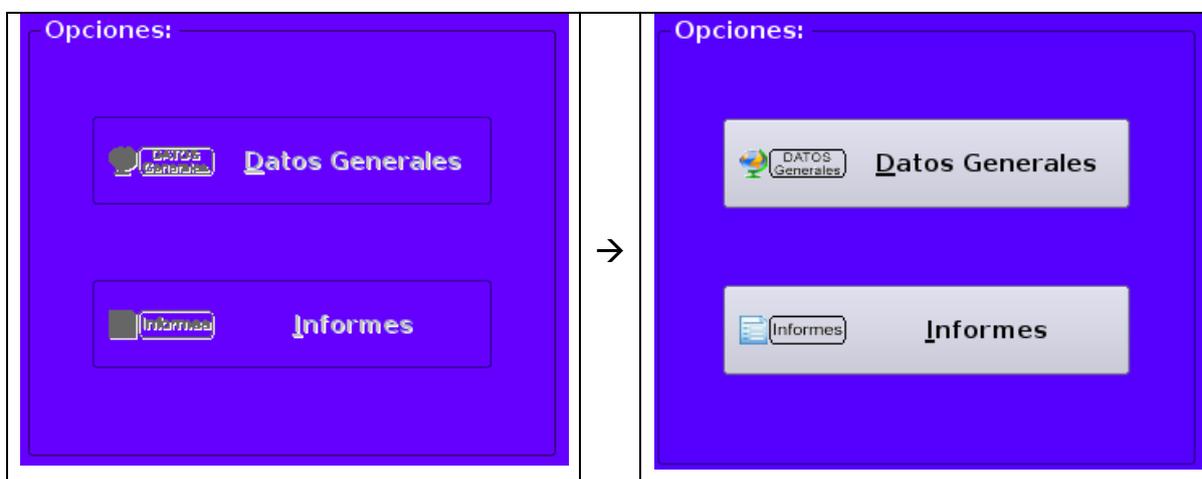


Figura 4.35. Opciones de Acceso a datos

Los Datos Generales así como los informes permitirán tener acceso a datos representativos del recorrido seleccionado.

La Pantalla de informes tendra las características que se muestran en la siguiente Figura:



Figura 4.36. Pantalla de Informes Gráficos



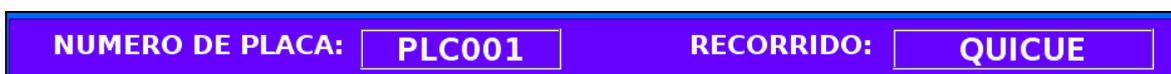
- El botón volver permitirá regresar a la pantalla principal para poder tener acceso a cualquiera de las opciones que la pantalla principal nos presenta.



- El botón imprimir nos dará la posibilidad de obtener la impresión del reporte del recorrido.



- El botón ver permitirá obtener una visión preliminar del reporte a ser impreso.



La Pantalla tendrá la Posibilidad de visualizar la Placa del vehículo seleccionado así como el recorrido que se va a analizar.

DATOS:

Velocidad Max.:

Velocidad Min.:

Hora de inicio del recorrido:

Hora de finalizacion del recorrido:

- Permitirá Obtener

valores relevantes del recorrido asi como Velocidad máxima, velocidad mínima, hora de inicio del recorrido y horma de finalización del recorrido.

La Pantalla del reporte brinda la posibilidad de visualizar mediante una tabla los puntos en los cuales la velocidad del recorrido rebaso los 70 Km/h, pudiendo obtener la ubicación y el tiempo exacto donde hubiere ocurrido.

Puntos en los que la velocidad es mayor a 70 Km/h

TIEMPO	VELOCIDAD	LONGITUD	LATITUD
161201	151	12158.1E	3723.1N
164204	254	12158.20	3723.2S
165205	155	12158.3E	3723.3N
173208	158	12158.40	3723.4S
175210	160	12158.50	3723.5S
182212	166	12158.60	3723.6S

Figura 4.37. Presentación de Velocidades mayores a 70Km/h

Se puede visualizar una grafica interactiva que mediante el deslizamiento del puntero del Mouse nos dará la posición y velocidad del punto en el que nos estamos deslizando, permitiendo de esta forma conocer los puntos críticos en la grafica de una manera sencilla.

CAPITULO V

ANÁLISIS E INSTALACIÓN DEL PROTOTIPO

5.1 INTRODUCCIÓN

El prototipo ha sido desarrollado de tal manera que sea modular y que permita instalar fácilmente en un vehículo para sus respectivas pruebas.

En éste capítulo se detallan los costos generados para el diseño e implementación del prototipo. Es conveniente tener en cuenta que los costos mayores para dispositivos que requieren investigación y desarrollo de ingeniería.

Se cuenta con un manual de usuario que ayudará a la instalación y funcionamiento correcto del prototipo en un vehículo.

5.2 DATOS TÉCNICOS DEL PROTOTIPO

5.2.1 Alimentación

Voltaje: 12.0 – 13.0 Vdc.

5.2.2 Consumo de corriente

Funcionamiento normal: 0,830 A o 830mA

Grabación de CD: 1,2 A

5.2.3 Tiempo de muestreo de datos

1 segundo, se graban todas las variables.

5.2.4 Transferencia de Datos a CD-RW

tiempo de prueba = 30 min

tiempo de lectura de la SD Card = 5 min

Bytes almacenados por segundo = 512

Cantidad total de bytes = $30 \text{ min} * \frac{60s}{\text{min}} * 512 \text{ bytes} / s = 921.600 \text{ bytes}$

Velocidad de transmisión = $\frac{921.600 \text{ bytes}}{5 \text{ min} * \frac{60s}{\text{min}}} = 3.072 \text{ bytes} / s \approx 3,1 \text{ Kbps}$

5.3 INSTALACIÓN DEL SISTEMA PROTOTIPO EN EL VEHICULO DE PRUEBA.

El prototipo ha sido instalado en un vehiculo Gran Vitara 4x4 año 2002



Figura 5.1 Vehículo utilizado para instalación del prototipo.

5.3.1 INSTALACIÓN DEL PROTOTIPO

5.3.1.1 Alimentación

El dispositivo ATAPI utiliza también 12V para su funcionamiento, y para mayor seguridad se trabaja con el regulador L7812 para este valor de voltaje.

El diseño de la fuente es basado en la potencia nominal que consume el prototipo, permitiendo también el diseño del disipador de calor.

En la siguiente tabla se muestra el consumo de potencia de cada uno de los dispositivos y el sistema en general:

Dispositivo / sección	Corriente (A)	Voltaje (V)	Potencia (Watt)
GPS ET-102	0,06	5	0,3
SD Card	0,075	3,3	0,25
PIC16F877A (Micro 1)	0,025	5	0,13
PIC16F877A (Micro 2)	0,025	5	0,13
Hitachi LCD	0,05	5	0,25
Max232	0,015	5	0,08
Circuito complementario	0,03	5	0,15
Total	0,28		1,2725

Tabla 5.1 Potencia consumida por el prototipo para la fuente de 5V

Dispositivo / sección	Corriente (A)	Voltaje (V)	Potencia (Watt)
Samsung CDRW 523252x (12V)	1,4	12	16,8
Samsung CDRW 523252x (5V)	0,6	5	3
Total	2		19,8

Tabla 5.2 Potencia consumida por el prototipo para la fuente de 12V

Diseño del protecciones

La principal protección es el fusible que será dimensionado tomando en cuenta que se tienen fuentes diferentes para el circuito del prototipo y el CD RW.

Fusible para el circuito de I prototipo:

$$fusible(mA) = 0,28 + 15\% = 0,322mA$$

Fusible para el CD RW

$$fusible(A) = 2 + 15\% = 2,3A$$

La fuente de voltaje que alimentará al sistema es tomada de la batería del automóvil, que tiene un voltaje de 12.7V.

El conector fue ubicado en la parte inferior de los asientos de atrás del vehículo como se puede ver en las fotografías.

5.3.1.2 Controlador y CD-RW

El CD-RW tiene que ser montado en forma horizontal o vertical como se muestra en la fotografía. Esto es de acuerdo a las especificaciones de este dispositivo.



Figura 5.2 Dispositivos del Prototipo.

Su alimentación es tomada del conector que es exclusivamente para el dispositivo ATAPI.



Figura 5.3 Conexiones del prototipo

5.3.1.3 Interfaz de usuario

Existen dos formas de conectar: la primera es a través del cable de interfaz que tiene una longitud de 60cm, pudiéndose colocar en cualquier lugar a la distancia especificada del cable.



Figura 5.4 Conexión de Interfaz de usuario con conector

La segunda manera es conectar directamente al controlador, para la aplicación desarrollada es la mejor opción.



Figura 5.5 Conexión de Interfaz de usuario directa

5.3.1.4 Sensor de Velocidad

El sensor de velocidad consta de un magneto, y se coloca en el vehículo de tal manera que el interruptor y el magneto en cada vuelta que da la rueda se pongan en línea generando un pulso.

Este puede ser ubicado en cualquiera de las ruedas del vehículo, tomando en consideración las características del diseño.

5.3.1.5 Antena de GPS

La antena GPS tiene un magneto que se adhiere al metal, que puede estar fijada hasta una velocidad de 180Km/h, superior a esto se debe asegurar la antena.

La mejor ubicación es en la parte superior externa del automóvil, principalmente para captar mejor la señal del Satélite.

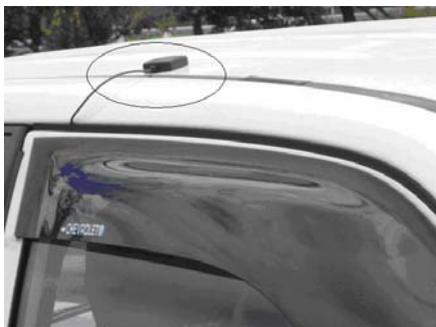


Figura 5.6 Ubicación de la antena del GPS

5.3.2 INICIO DEL SISTEMA Y ADQUISICIÓN DE DATOS

- a) Alimentamos el sistema a través del switch de dos posiciones.
- b) Se 5 minutos (*tiempo de espera*) para que el módulo GPS obtenga datos correctos del satélite.
- c) Luego se presiona **Start** para que empiece a recibir y grabar datos.



Figura 5.7 Pantallas de Interfaz de Usuario

- d) Para la prueba se realizó un recorrido de 30 minutos, para posteriormente enviar a grabar al CD.

5.3.3 GRABACIÓN EN CD

- a) Se presiona **Select**, y se visualiza el mensaje de **Grabar CD?**, para lo cuál se presiona nuevamente **Start**.

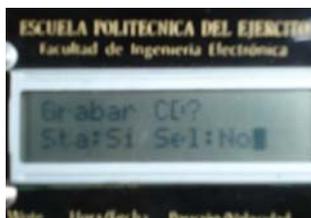


Figura 5.8 Pantalla de usuario para Grabar CD

- b) En ese instante se visualiza en la pantalla la transferencia de los datos, y se debe esperar hasta que toda la información sea receptada por el CD-RW.
- c) La grabación dura alrededor de 30 a 45 minutos.



Figura 5.9 Pantalla de transferencia de Datos

5.3.4 TRANSFERENCIA DE DATOS A LA PC (Personal Computer)

- a) Se ingresa a Linux/Ubuntu, e inicie la sesión como súper usuario o root.
- b) Se coloca el CD en el CD-RW, se debe esperar hasta que sea reconocido por la PC.
- c) Abrir el Terminal y ejecutar el programa de lectura de CD. Este paso depende del programa que se haya instalado tomando en cuenta la dirección y configuración del CD-RW en la PC. Y tenemos la siguiente pantalla que indica una correcta lectura del CD.
- d) Este programa genera un archivo datos1.txt, en donde se encuentra las variables sensadas. Y tiene el siguiente formato:

5.1.1 VISUALIZACIÓN Y MANIPULACIÓN DE VARIABLES

Las variables que el sistema capta y graba en el cd para generar el archivo datos.txt nos da la posibilidad de obtener los datos del recorrido que son:

- Hora
- Fecha
- Longitud
- Latitud
- Velocidad

El programa SSV 1.0 nos brinda la posibilidad de almacenar en arreglos cada una de las variables por separado.

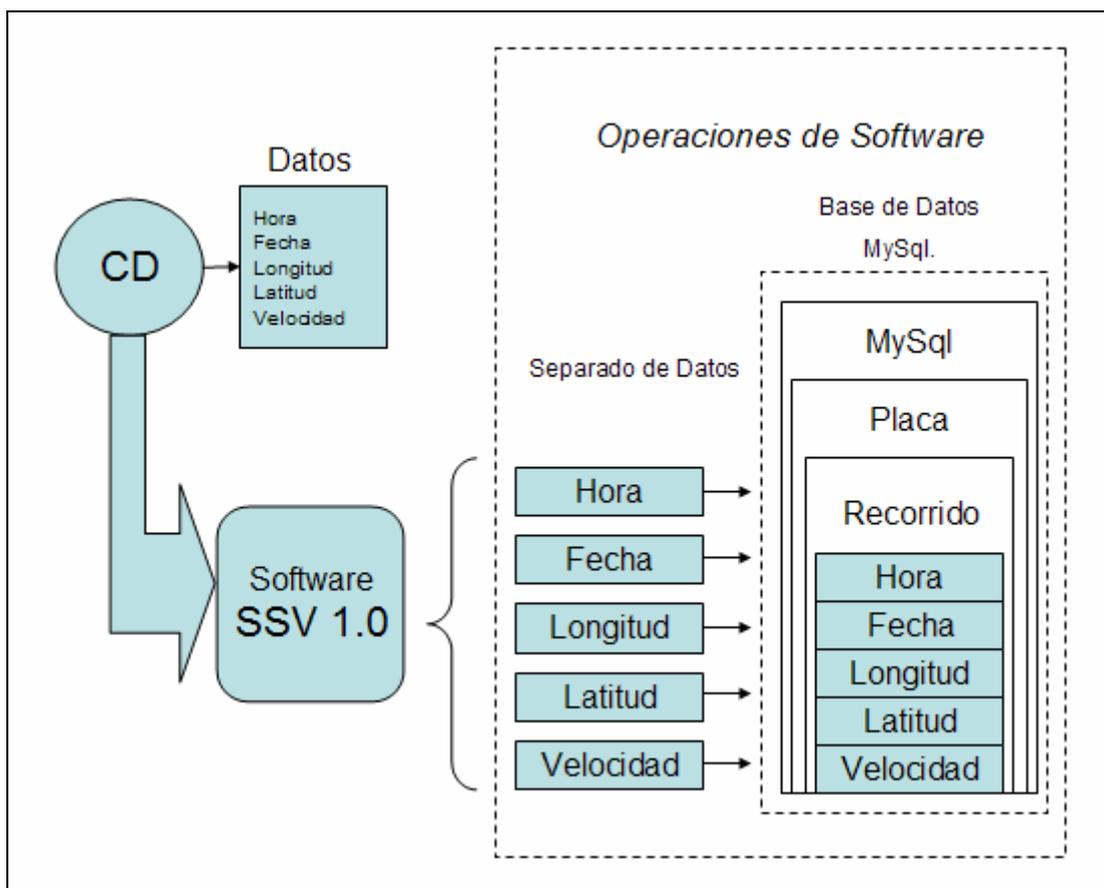


Figura 5.10 Esquema de variables y almacenamiento en el gestor de Base de Datos MySQL.

5.2 MANUAL DE USUARIO

En el manual de usuario se encuentra descrito todos los requisitos de instalación tanto de la parte física, así como el programa que manejará este sistema.

El manual se encuentra en el Anexo 9.

5.3 ANÁLISIS DE COSTOS

Los costos son directos, y son los necesarios para implementar un prototipo, sin contar los costos de instalación. A continuación se muestra en detalle los costos.

COSTO DIRECTO DEL SISTEMA			
DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO	PRECIO TOTAL
Módulo GPS ET-102	1	95,00	95,00
LCD hitachi 16*2	1	9,00	9,00
PIC16F877A	2	7,00	14,00
SD Card 128MB	1	25,00	25,00
PCB (placa) 11*11 cm	1	38,00	38,00
Pulsadores	5	0,70	3,50
Resistencias	18	0,15	2,70
Caja de controlador	1	10,00	10,00
Caja de interfaz	1	10,00	10,00
Contacto magnético	1	2,75	2,75
Cable serial	1	1,00	1,00
Cable de Interfaz	1	1,50	1,50
CD RW	1	30,00	30,00
COSTO			242,45

Tabla 5.3 Costo directo del prototipo

Posibles mejoras al prototipo.

- Se podría unir el programa Atapisoft 1.0 al Software SSV 1.0.
- Al tener la licencia profesional del compilador CC5x se podría trabajar con un solo microcontrolador PIC 16F877A, debido a que permite compilar al 100%, ya que la versión Demo solo permite compilar al 50%.

- Se puede reemplazar el microcontrolador con uno de mayor capacidad o adicionar memoria rom a prototipo.
- Se podrían adaptar más sensores, siempre y cuando se acondicione a señales analógicas, ya que los pines con estas características se encuentran habilitados. Y se debería realizar la programación para la adquisición de estos datos.
- La utilización de un GPS diferencial serviría para tener mayor precisión en la ubicación, se podría adaptar un GPS Gramin.
- La placa PCB se debería ser hecha en un mejor material y con mayor tecnología con la que se tiene aquí en Quito.
- Realizar la caja contra choques y que filtre todos los ruidos posibles.
- Mejorar la visualización del reporte en SSV 1.0, mediante la librería GBreport.

CAPITULO VI

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- El desarrollo de este prototipo con el fin de proveer datos para la investigación de accidentes de tránsito ha generado la posibilidad del desarrollo de nuevas etapas para un control integral.
- Los mensajes RCM que envía el satélite y recibidos por el módulo GPS ET-102 a una frecuencia de 1Hz, es suficiente y necesario para el diseño e implementación del prototipo del sistema de seguridad vehicular., teniendo una precisión de ± 3 metros en la ubicación.
- La comunicación con la SD Card a través del protocolo SPI es de acceso lento, por lo que la velocidad de transmisión de los datos almacenados al CD-RW llega a 3.1kbps.
- El estándar ATA/ATAPI, es muy estricta en cuanto a tiempos de espera de transferencia de comandos para el acceso a los registros del dispositivo. Dicho acceso solo se permite a la configuración de master, cuando se trabaja externamente sin la PC (Personal Computer).

- La única manera de grabar información en un CD con un modulo externo es hacerlo por direcciones. Para un CD de 700MB se tiene 341.796 direcciones con un tamaño estándar de 2048 kb. Para grabar datos se inicia desde la dirección 0.
- Las SD Cards son dispositivos de almacenamiento de bajo costo desde \$15 (128MB) – \$75 (4GB) y de mucha oferta en el Mercado ecuatoriano, de forma que se utiliza como memoria de almacenamiento para diferentes equipos como cámaras, celulares, entre otros. La simplicidad del protocolo SPI de la SD card y la flexibilidad en la interfaz con estos dispositivos hace ideal para el acceso con los microcontroladores (PIC16F877A).
- Éste prototipo utiliza una placa de circuito impreso (PCB) de baja calidad hecha en la ciudad de Quito. No se encontró sitios o lugares donde se realicen PCB de mejor calidad. Este factor causa condiciones fuera de lo normal en el funcionamiento
- Linux permite manipular todos los puertos de comunicación, simplemente colocando la dirección del mismo. Es importante pedir permiso antes de enviar o recibir datos de/hacia los dispositivos. Linux también maneja librerías similares a Borland C++ para Windows, excepto en Dos.H (windows) porque en linux es ioper.h
- Atapisoft 1.0 utiliza los compiladores básicos de linux son el gcc y g++, los cuales permiten el desarrollo del software para la lectura de datos desde el CD. Para la ejecución el Atapisoft no es necesario tener el compilador g++, ya que es un ejecutable que puede ser instalado en el sistema Ubuntu a partir de la versión 6.10
- El microcontrolador PIC6F877A, tiene las características necesarias para cumplir con los requisitos del prototipo, por lo que se ha utilizado en un 100% las funciones del microcontrolador.

- El compilador Cc5x que es utilizado para el PIC6F877A, optimiza en un 40% el código de programación a diferencia del CCS y Piclite. Una de las dificultades es la compilación únicamente del 50% en la versión free demo.
- La estructura de un vehiculo hace que genere alto ruido blanco, afectando a los circuitos electrónicos que son sensibles a este factor. Causando errores en el sistema cuando este no esta bien aislado y con filtros que eliminen el ruido.
- La utilización del lenguaje de programación GAMBAS con GPL (Licencia Publica General) posibilita el acceso a recursos necesarios para el desarrollo del Software de interpretación de datos generados por el prototipo S.S.V.
- La optimización de recursos tanto en codificación como en manejo de la interfaz gráfica que brinda el compilador de Basic que posee GAMBAS hace posible que los requerimientos del sistema para el desarrollo de la aplicación sean mínimos.
- La falta de información sobre el desarrollo de nuevos componentes que faciliten el entorno gráfico e incorporen propiedades al software, en cierta forma no permite mejorar la visualización.
- El impulso de librerías para aplicaciones específicas a hecho posible la conexión de Gambas con Bases de Datos.
- El manejo del gestor de Base de Datos MYSQL con su aplicación para Linux brinda la posibilidad de almacenamiento de información, permitiéndonos acceder a la misma de forma rápida.

RECOMENDACIONES

- El desarrollo de material informativo sobre el manejo del Lenguaje de programación GAMBAS facilitaría el desarrollo de nuevas aplicaciones, debido a que la información acerca de este lenguaje de programación es mínima.
- Para la instalación y ejecución del Software SSV 1.0 y Atapisoft 1.0 se recomienda trabajar desde la sesión de root en LINUX, con esto se evitará conflictos con permisos para el acceso a bases de datos y puerto IDE/Atapi.
- La Facultad de Electrónica tiene que afianzar conocimientos no únicamente con los microcontroladores PIC16F877A, sino también de la familia Atmel los cuales son los más utilizados en el mercado.
- La Facultad de Electrónica adquiera licencias de los compiladores para el trabajo con la casa Microchip y Atmel, por cuanto son necesarios para el desarrollo e implementación de proyectos donde se utilizan los microcontroladores.
- Configurar los mensajes que el módulo GPS envía hacia el controlador, para que el funcionamiento del prototipo sea normal.

BIBLIOGRAFÍA

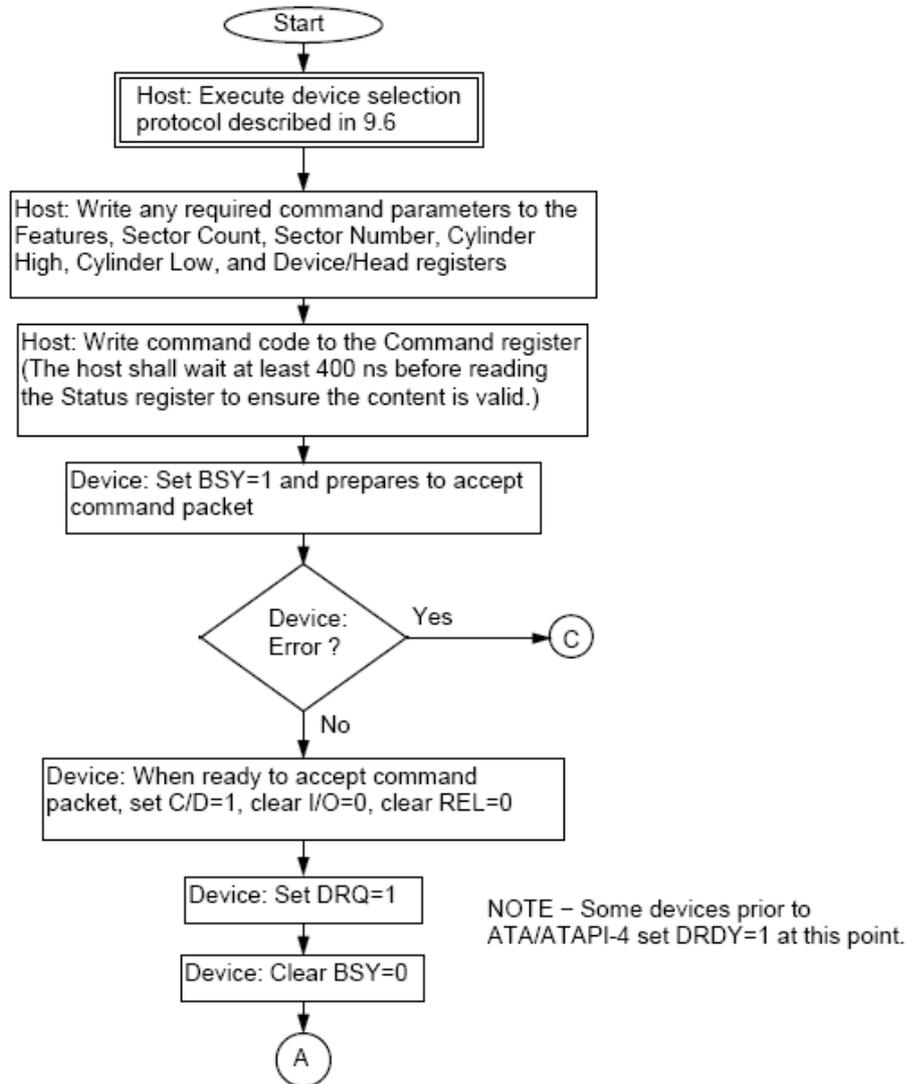
1. James A. Blackburn, Modern Instrumentation for Scientists and Engineers (Springer, 2003).
2. Jhon Park, Data communications for instrumentation and Control (Newnes,2003).
3. Paul Dubois, MySQL (Primera edición, AnayaMultimedia, España, 2005)
4. http://www.inec.gov.ec/interna.asp?inc=enc_tabla&idTabla=227, Accidentes de Tránsito a Nivel Nacional (años 2003-2004), febrero 2006.
5. <http://www.autotrackedcuador.com/index.php?section=10>, Sistema de Control Satelital Vehicular, febrero 2005.
6. <http://www.homepages.mty.items.mx/disclaimer.html>, Comunicación con la tarjeta SD Card (Secury Digital)
7. <http://sdcard.org/>, SD Card Association
8. <http://www.usglobalsat.com/>, ET-102 GPS BOARD OVERVIEW
9. <http://www.sirf.com/>, NMEA and SIRF protocol manual reference, SIRF demo Software.
10. <http://www.tolaemon.com/otros/mmc card.htm>, Las tarjetas MultiMedia Card, julio 2004

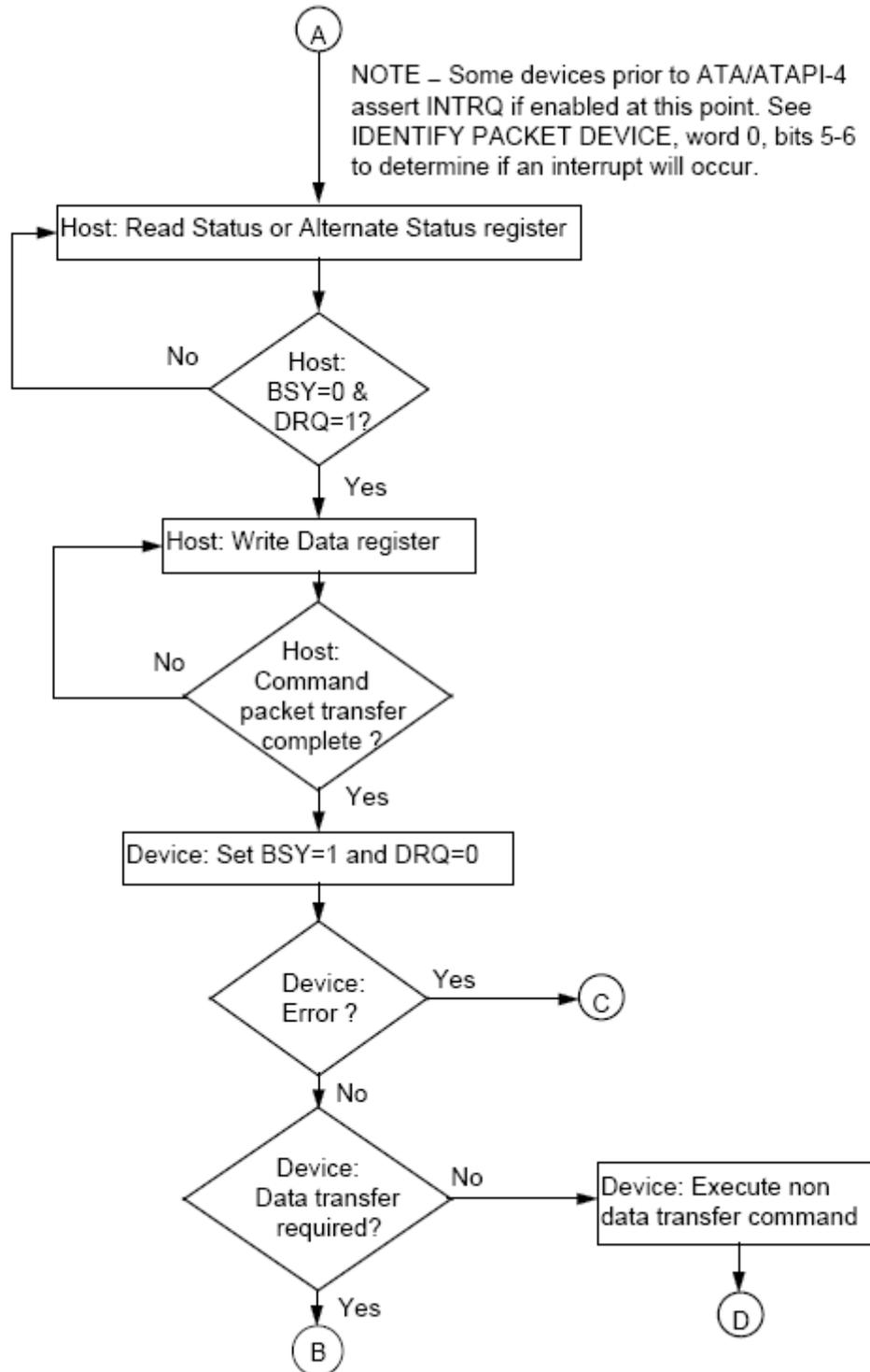
11. http://picmania.garcia-cuervo.com/Proyectos_AUX_LCD.htm, Control y Programación del Display LCD, noviembre 2006
12. <http://www.cdei.upc.es/>, Centro de Diseño de Equipos Industriales, marzo 2004.
13. <http://www.cc5x.de/MMC/files/MMC.zip>, Programación para microcontroladores
14. <http://www.brushelectronics.com/>, FAT32 code for MultiMedia Cards, agosto 2005
15. <http://www.gpsworld.com>, características del GPS, marzo 2006
16. <http://www.astroradio.com/t/tenda/index.html>, Mundo GPS, mayo 2004
17. ftp://ftp.pct.edu/pub/shared/mpx/ide_tech.txt, CD-ROM ATAPI interfacing
18. <http://www.ata-atapi.com/>, Driver C source code, octubre 2005
19. www.t13.org, "AT Attachment with Packet Interface - 6 (ATA/ATAPI-6)", pp. 415, 2000.
20. www.t13.org, Small Form Factor Committee, "ATA Packet Interface for CD-ROMs SFF-8020i", pp. 224, 1996.
21. http://en.wikipedia.org/wiki/C_programming_language, C programming language, abril 2006.
22. <http://www.symbios.com/x3t10>, MMC commands and SCSI Primary commands
23. <ftp://ftp.t10.org/t10/drafts/spc/spc-r11a.pdf>, SCSI commands
24. <http://come.to/eide>, IDE interface
25. <http://corsa.mforos.com/780631/4645463-instalar-cuentarevoluciones-corsa-b/>, Instalación de un cuenta revoluciones

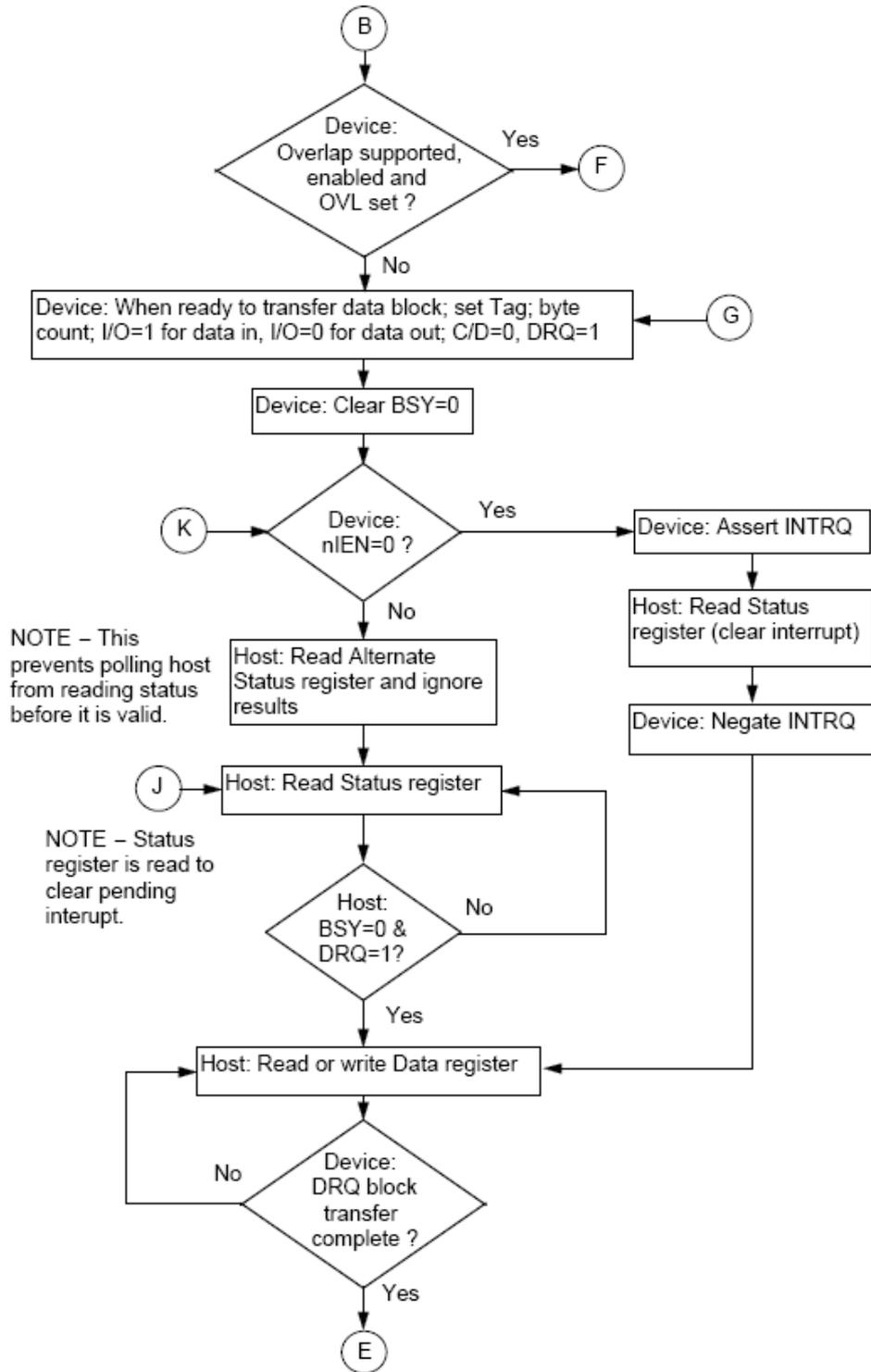
26. <http://www.el4x4.com/spa/item/85.html>, Instalación de un control de velocidad.

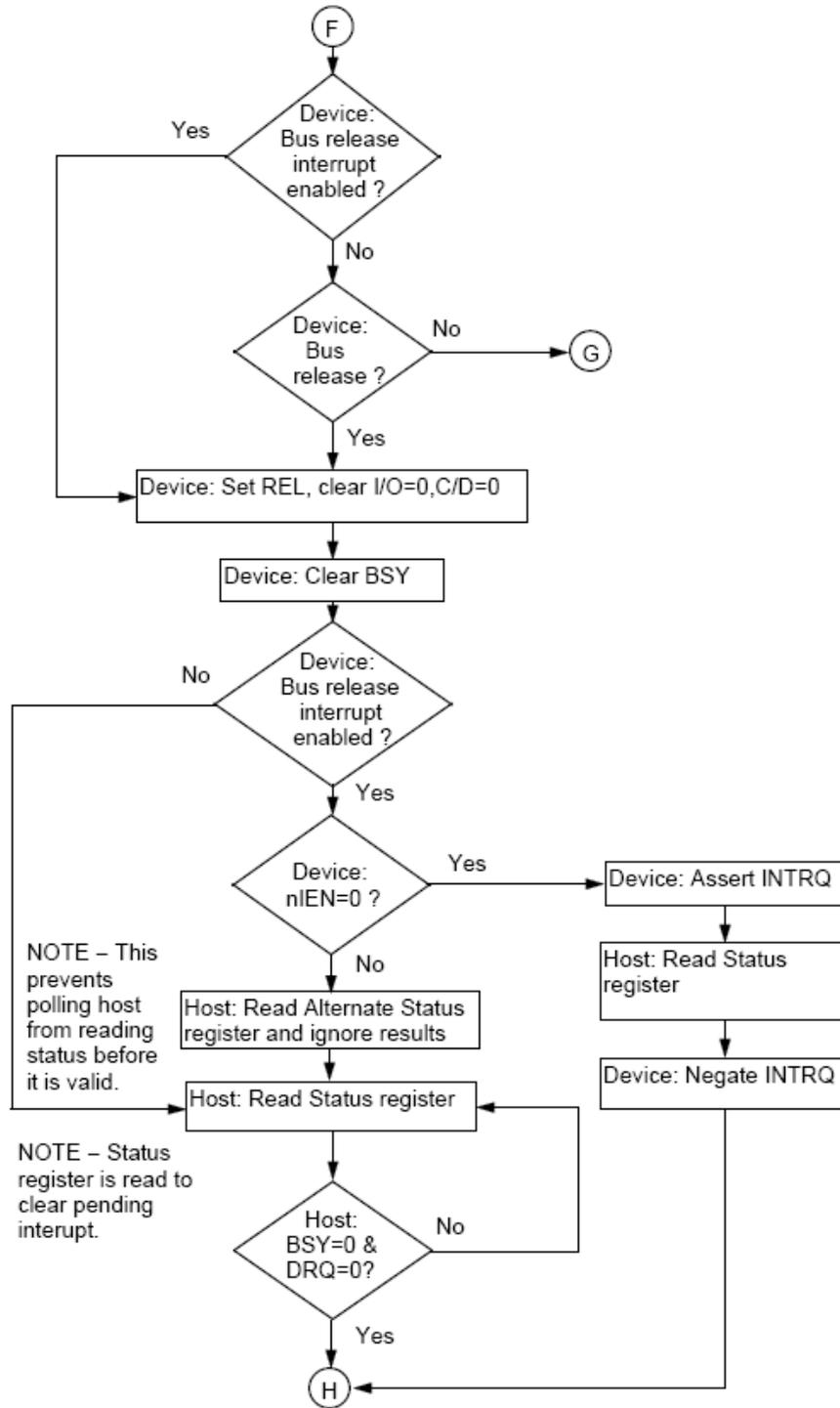
ANEXOS

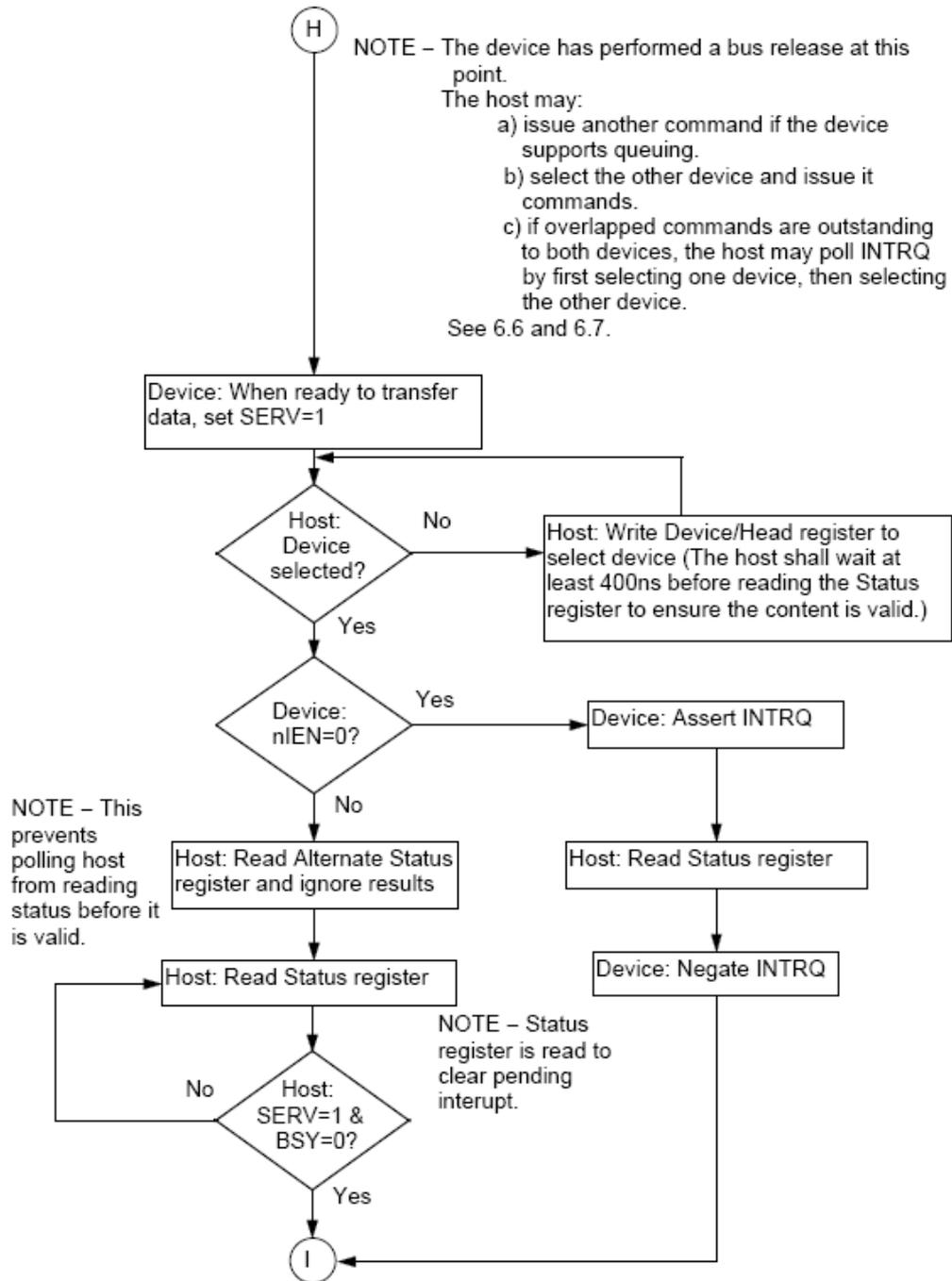
8.1 Anexo 1. Diagrama de flujo del envío del packet command al dispositivo

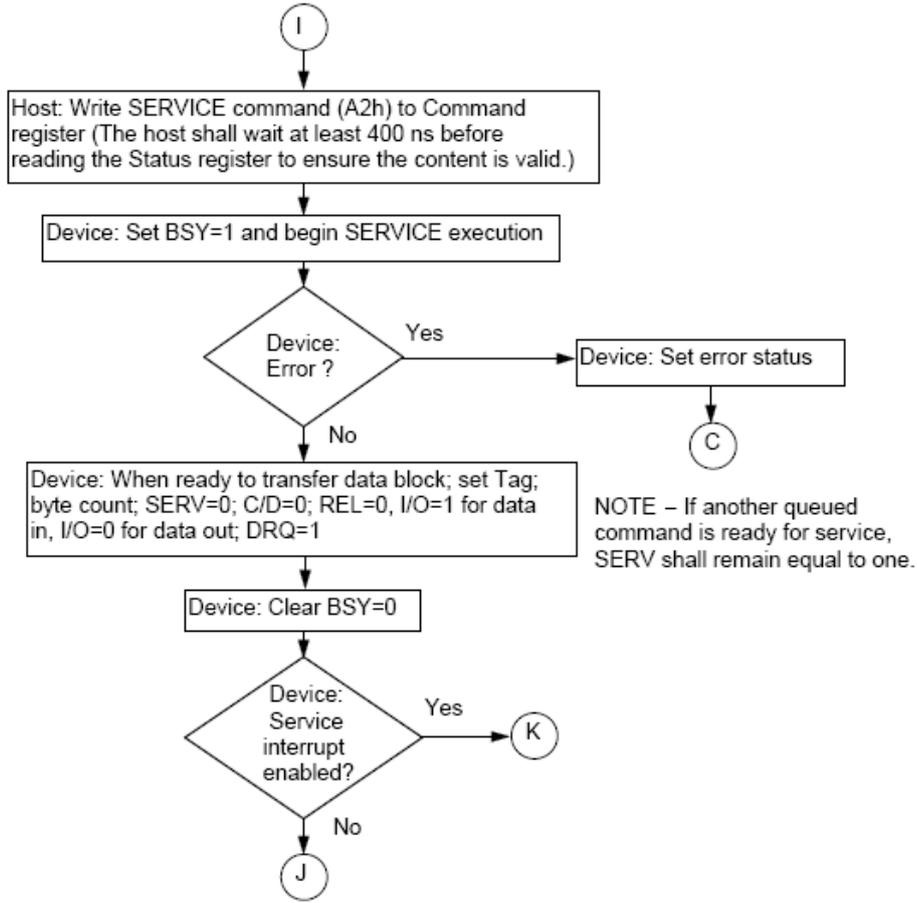




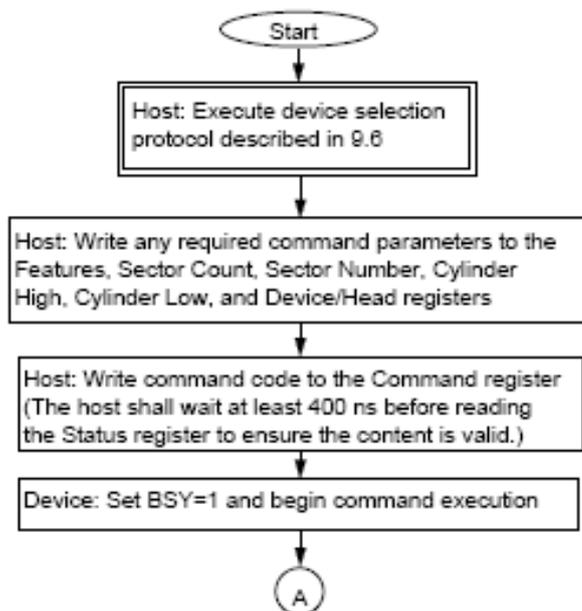


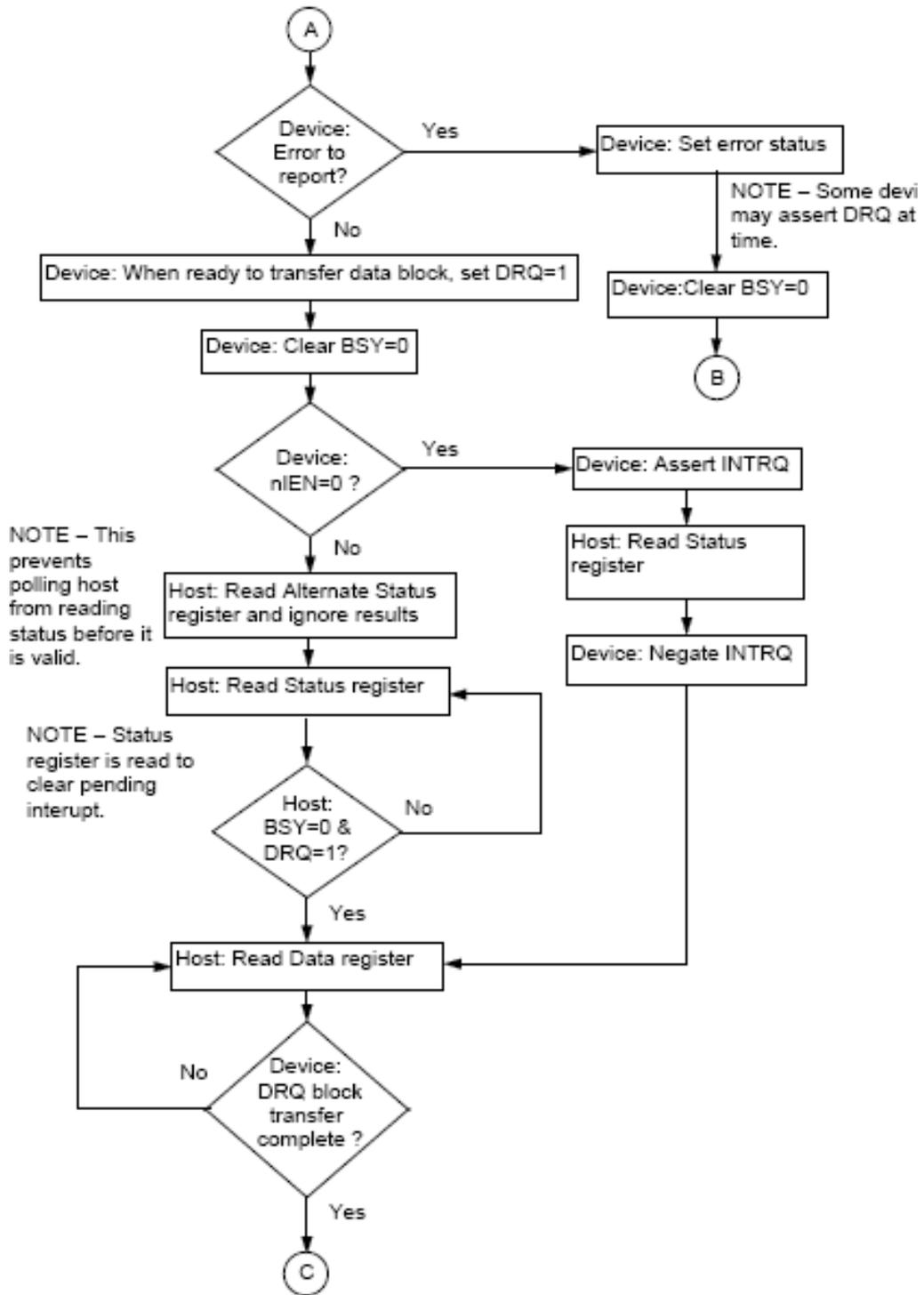


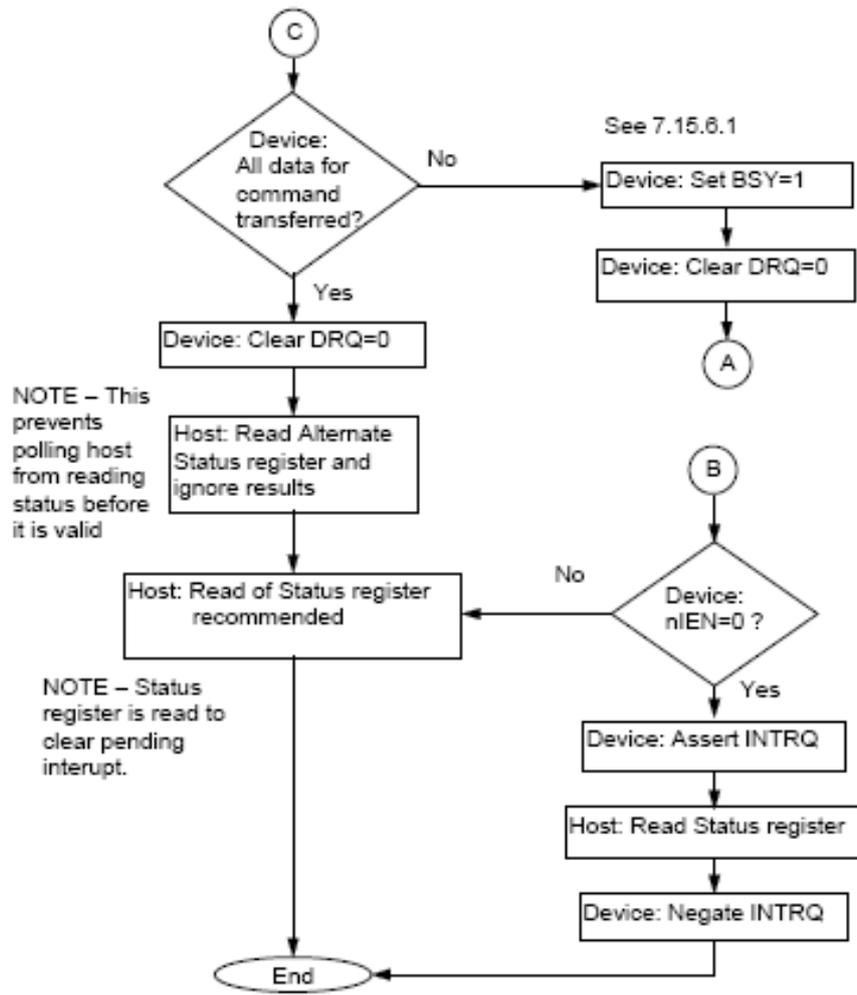




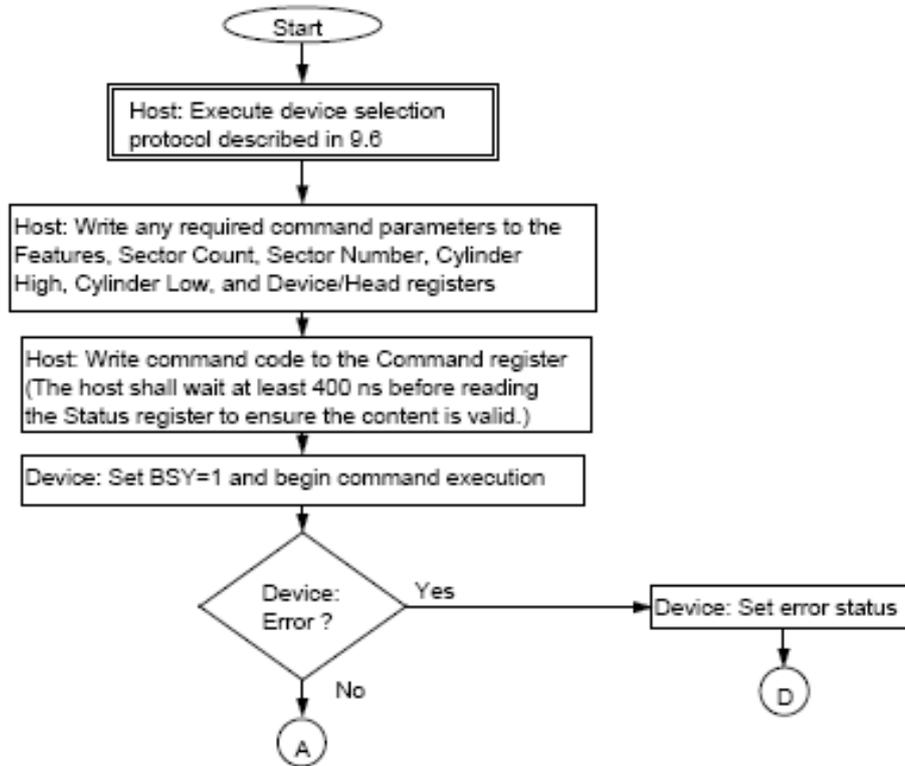
8.2 Anexo 2. Diagrama de flujo de comandos que requieren leer datos del dispositivo

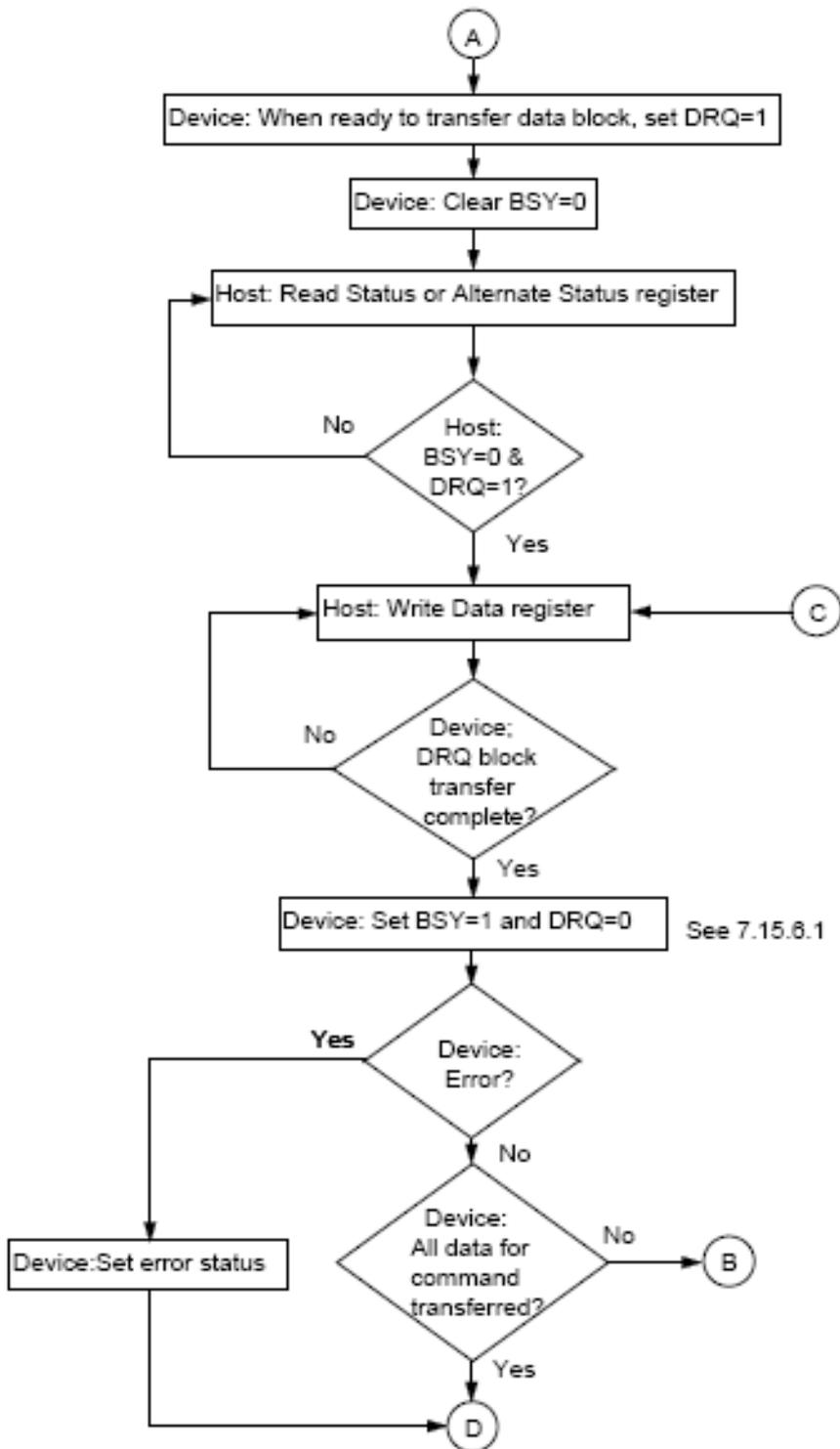


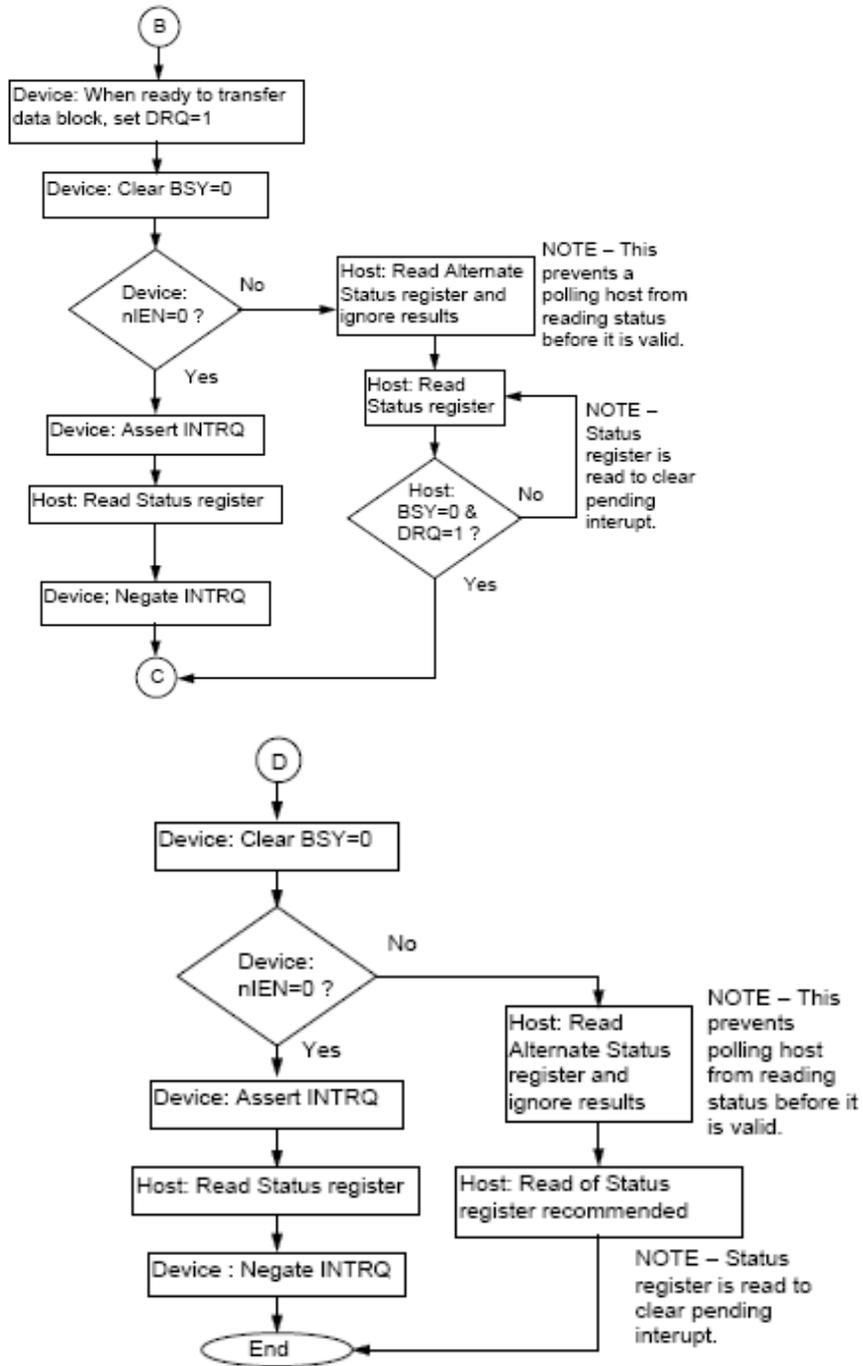




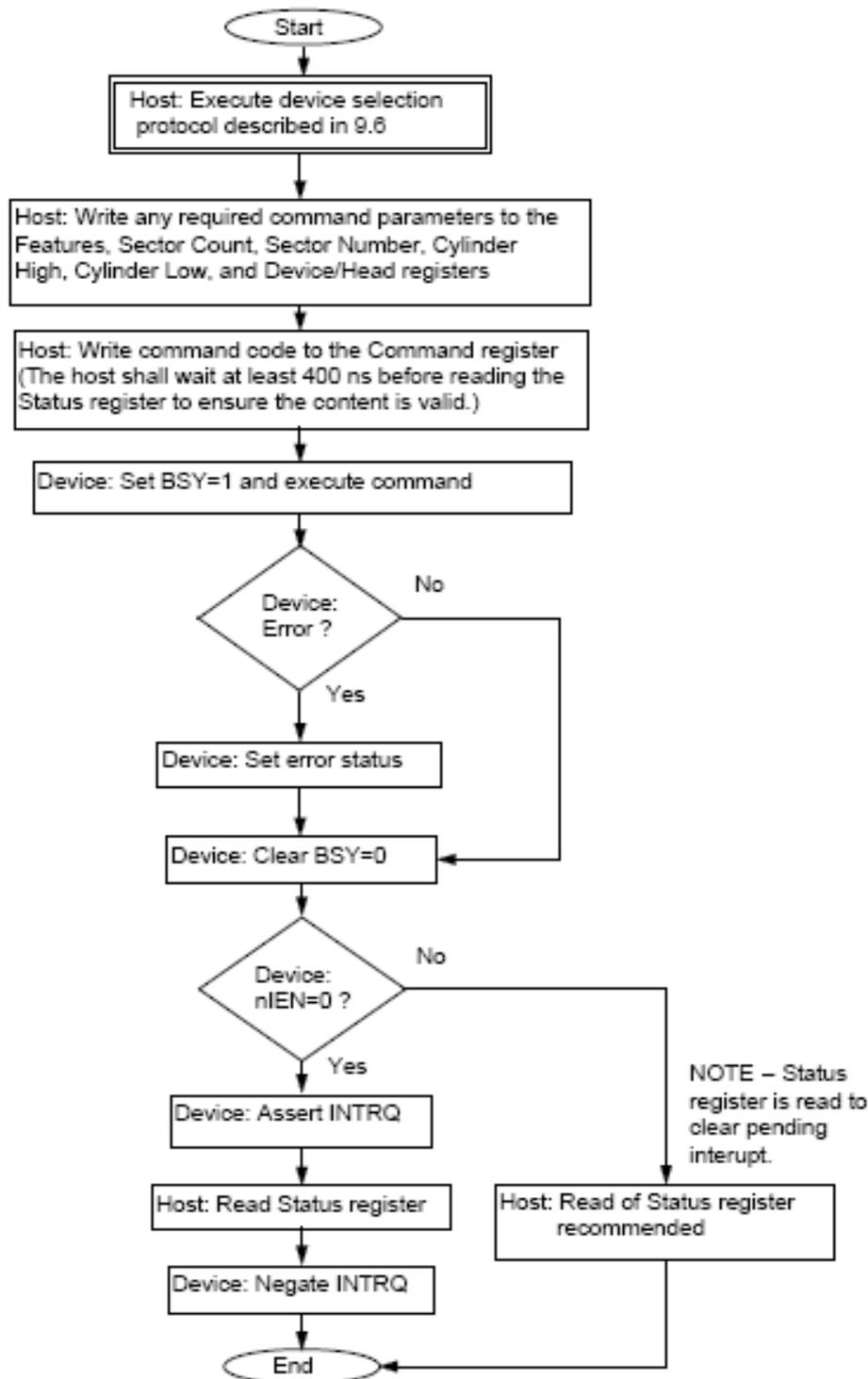
8.3 Anexo 3. Diagrama de flujo de comandos que requieren escribir datos al dispositivo







8.4 Anexo 4. Diagrama de flujo de comandos que no requieren datos



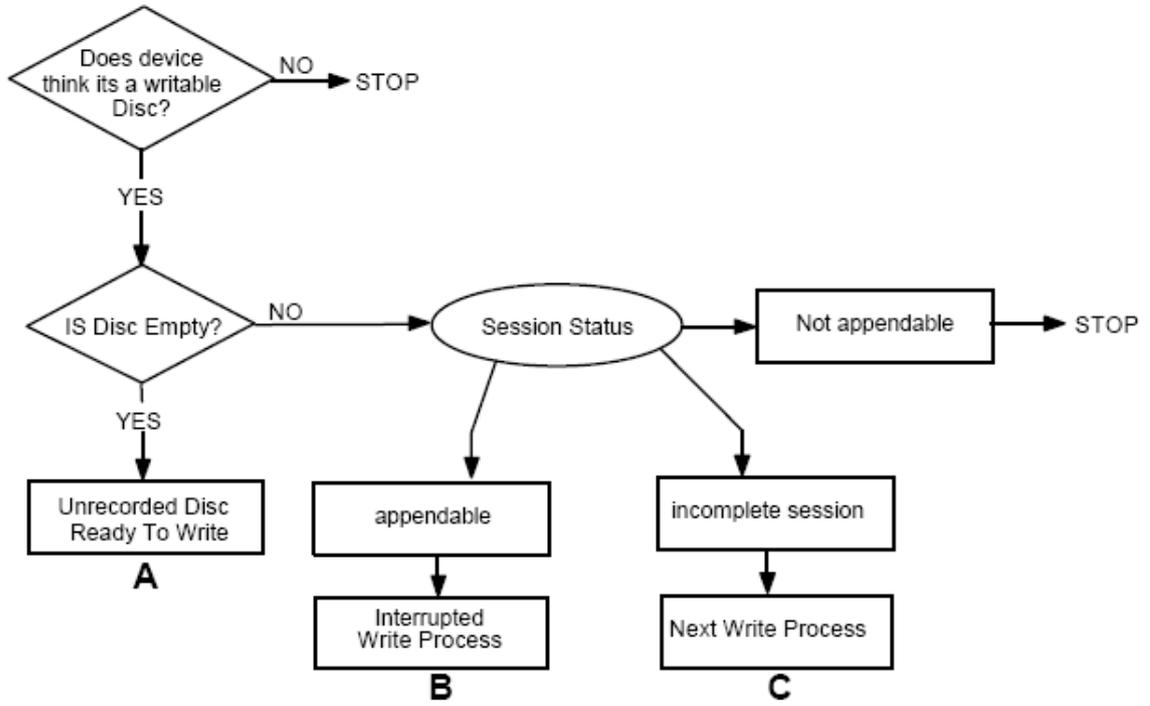
8.5 Anexo 5. Lista de ATA Commands

ATA Commands	Op-Code
ATAPI Soft Reset	08h
Check Power Mode	E5h
Execute Device Diagnostics	90h
Identify Device	ECh
Idle Immediate	E1h
ATAPI Packet Command	A0h
ATAPI Identify Device	A1h
Read Sectors (S) (w/ retry)	20h
Read Sectors (S) (w/o retry)	21h
Set Features	EFh
Sleep	E6h
Standby Immediate	E0h

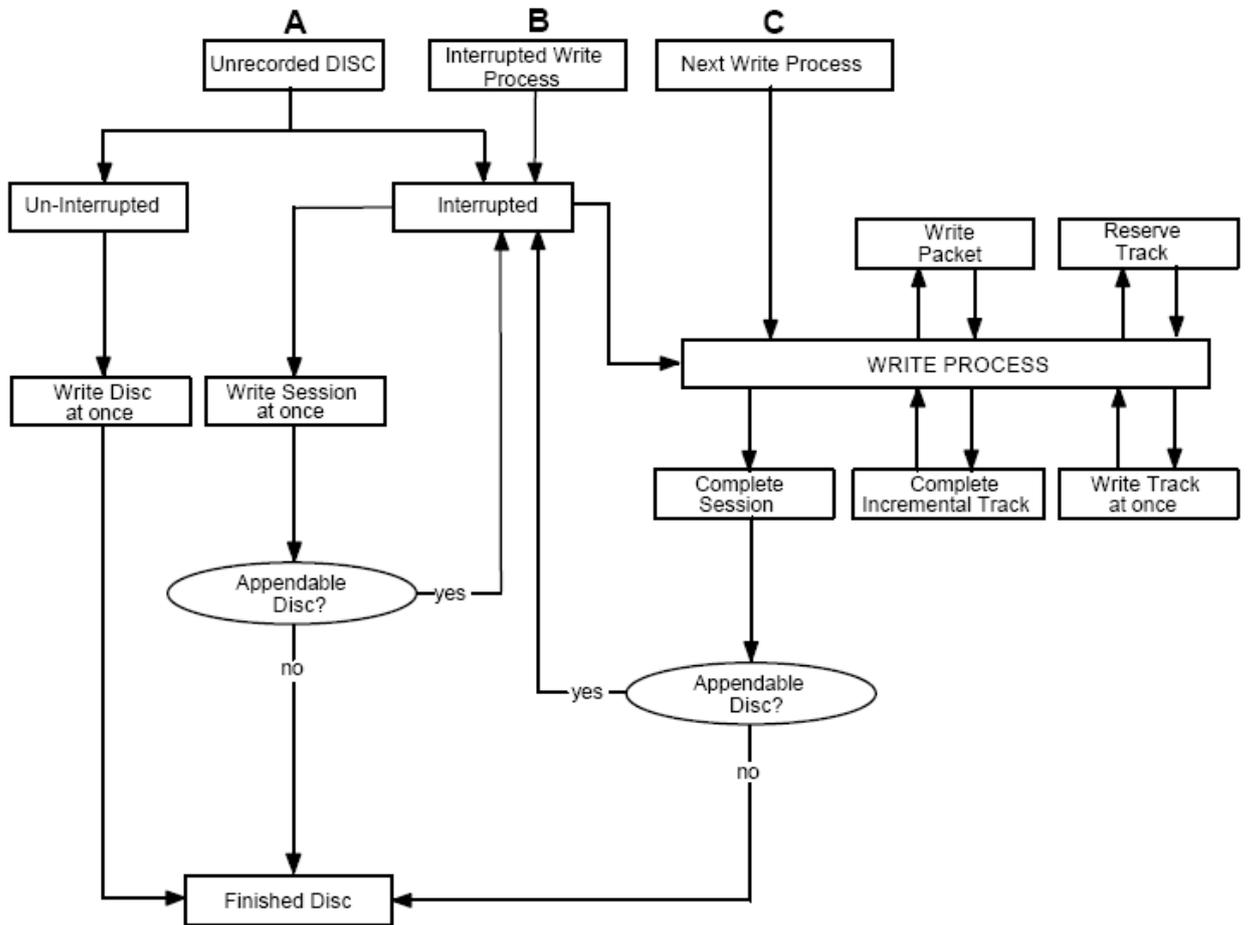
8.6 Anexo 6. CDROM ATAPI packet command

ATAPI Packet Commands	Op-Code
FLASH CACHE	35h
FORMAT UNIT	04h
GET CONFIGURATION	46h
GET EVENT STATUS NOTIFICATION	4Ah
GET PERFORMANCE (TBD)	ACh
INQUIRY	12h
MECHANISM STATUS	BDh
MODE SELECT (10)	55h
MOSE SENSE (10)	5Ah
PAUSE / RESUME	4Bh
PLAY AUDIO (10)	45h
PLAY AUDIO (12)	A5h
PLAY AUDIO MSF	47h
PLAY TRACK RELATIVE (10)	49h
PLAY TRACK RELATIVE (12)	A9h
PREVENT ALLOW MEDIUM REMOVAL	1Eh
READ (10)	28h
READ (12)	A8h
READ CAPACITY	25h
READ CD	BEh
READ CD MSF	B9h
READ DISC INFORMATION	51h
READ DVD STRUCTURE	ADh
READ FORMAT CAPACITIES	23h
READ HEADER	44h
READ SUB-CHANNEL	42h
READ TOC / PMA / ATIP	43h
READ TRACK / RZONE INFORMATION	52h
REASSIGN BLOCKS	07h
REPORT KEY	A4h
REQUEST SENSE	03h
REZERO UNIT	01h
SCAN	BAh
SEEK (10)	2Bh
SEND EVENT (TBD)	A2h
SEND KEY	A3h
SET CD SPEED	BBh
SET READ AHEAD	A7h
SET STREAMING (TBD)	B6h
START STOP UNIT	1Bh
STOP PLAY / SCAN	4Eh
SYNCHRONIZE CACHE	35h
TEST UNIT READY	00h
VERIFY	2Fh
VERIFY (12)	AFh
WRITE (10)	2Ah
WRITE (12)	AAh
WRITE AND VERIFY (10)	2Eh
WRITE AND VERIFY (12)	A Eh

8.7 Anexo 7. Reconocimiento del estado de un CD



8.8 Anexo 8. Proceso de Escritura



8.9 Anexo 9. Manual de Usuario