

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

“Almacenamiento masivo de sentencias NMEA del módulo GPS en memorias MMC/SD para su utilización posterior en mapas de posicionamiento”

TESINA DE SEMINARIO

PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

PRESENTADO POR:

Ricardo Xavier Gutiérrez Cevallos

Astrid Angie Zambrano Castillo

GUAYAQUIL – ECUADOR

2010

AGRADECIMIENTO

Ante todo a Dios por permitir que culmine con éxito esta etapa de mi vida, a mis padres por guiarme y brindarme la oportunidad de una excelente educación, mi abuela por estar siempre a mi lado, a mis hermanos y a mi tía, quienes siempre me dieron fuerzas para continuar.

Un agradecimiento especial a la Madre Dolorosa y al Niño Divino, pues siempre me iluminaron para alcanzar mis objetivos.

Ricardo Gutiérrez Cevallos

DEDICATORIA

Dedico el presente trabajo a mi familia, en especial a la Ing. Angela Cevallos Ponce, quien no solo es mi madre sino la persona más importante de mi vida; cuyo camino que ha forjado me ha servido como pauta para soñar, alcanzar y nunca desmayar en todo aquello que me proponga en la vida.

Dedicatoria especial a mis cachorros, quienes son símbolos de mi felicidad.

Ricardo Gutiérrez Cevallos

AGRADECIMIENTO

En primer lugar a Dios por darme la fortaleza de continuar cada día, en segundo lugar a mis queridos padres por lo todo su apoyo y amor incondicionalmente.

A mis profesores, amigos compañeros, y a todos aquellos que han contribuido a obtener esta importante meta profesional.

Astrid Zambrano

DEDICATORIA

Le dedico el presente trabajo a mi madre Lcda. Martha Castillo y a mi padre Ing. Pedro Zambrano, quienes no solo han sido el pilar fundamental sino también la fuente de inspiración y ganas para seguir adelante.

A la ESPOL, y a la FIEC como una demostración de gratitud por su excelencia académica.

Astrid Zambrano

TRIBUNAL DE SUSTENTACIÓN

Ing. Carlos Enrique Valdivieso A.
PROFESOR DEL SEMINARIO DE GRADUACIÓN

Ing. Hugo Villavicencio V.
DELEGADO DEL DECANO

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta Tesina, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”.

(Reglamento de exámenes y títulos profesionales de la ESPOL).

Ricardo Xavier Gutiérrez Cevallos

Astrid Angie Zambrano Castillo

RESUMEN

Actualmente el Sistema de Posicionamiento Global GPS es una de las principales tecnologías utilizadas en diversas aplicaciones para fines de orientación y sistemas de posicionamiento.

En este trabajo se describe el diseño e implementación de un sistema que utiliza como herramientas funcionales un receptor GPS, una tarjeta ET-MINI SD/MMC y un microcontrolador. Los dispositivos en conjunto proveen la capacidad de almacenamiento masivo de cadenas NMEA.

Se aprovecha la tarjeta de desarrollo SmartGPS como una herramienta educativa, pues provee facilidades al usuario en su modo de funcionamiento; de igual manera la tarjeta ET-MINI SD/MMC resulta beneficiosa como medio físico para comunicar el microcontrolador y la memoria de tipo MMC o SD.

En el desarrollo del proyecto se hace uso de la tarjeta de entrenamiento DSPIC PRO4 que provee un manejo sencillo para la interacción con las memorias MMC/SD en el almacenamiento de datos.

ABREVIATURAS

GPS Global Positioning System (Sistema de Posicionamiento Global)

I2C (Inter Integrated Circuit) Circuito Inter Integrado

NMEA National Marine Electronics Association (Asociación Nacional de Electrónica Marina)

DC Direct Current (Corriente directa)

DIP Dual in-line package (Empaque de doble línea)

ESPOL Escuela Superior Politécnica del Litoral

FIEC Facultad de Ingeniería de Electricidad y Computación

PWM (Pulse Width Modulation) Modulación por ancho de pulsos

RAM Random-access Memory (Memoria de Acceso Aleatorio)

RCIF (Receive Interrupt Flag bit) Bit de Bandera de Recepción de Interrupción

RS232 Recommended Standard 232 (Estándar Recomendado 232)

SPI (Serial Peripheral Interface) Interfaz Periférica Serial

USART Universal Synchronous Asynchronous Receiver Transmitter (Receptor Transmisor Síncrono Asíncrono Universal)

MMC Multi Media Card (Tarjeta Multimedia)

SD Secure Digital (Tipo de memoria de almacenamiento)

INDICE GENERAL

AGRADECIMIENTO	II
DEDICATORIA	III
AGRADECIMIENTO	IV
DEDICATORIA	V
TRIBUNAL DE SUSTENTACION	VI
DECLARACIÓN EXPRESA	VII
RESUMEN	VIII
ABREVIATURAS	IX
ÍNDICE GENERAL.....	X
ÍNDICE DE FIGURAS.....	XIII
ÍNDICE DE TABLAS	XV
INTRODUCCIÓN	XVI
CAPITULO 1	1
1 Descripción General del Sistema.....	1
1.1 Alcance y Limitaciones del Proyecto	1
1.1.1 Descripción del Proyecto	1
1.1.2 Estrategia Implementada.....	2
1.1.3 Limitaciones del Proyecto.....	3
1.2 Análisis de Soluciones Existentes en el Mercado.....	4
1.3 Aplicaciones	5
CAPITULO 2.....	7
2 Fundamentación Teórica.....	7
2.1 ¿Qué es un GPS y cómo funciona?.....	7
2.2 El Protocolo NMEA	9
2.3 Modos de Transmisión de Datos.....	12
2.4 Comunicación Serial	13
2.4.1 Tipos de Comunicación Serial	14

2.4.1.1	El Protocolo SPI	15
2.4.1.2	La Norma RS-232	17
2.5	Sistema de Archivos	19
2.6	Conceptos Básicos de Programación	19
2.7	Herramientas de software	20
2.7.1	MikroBasic Pro for PIC.	21
2.7.2	MikroBasic Pro for dsPIC.....	21
2.7.3	Proteus	21
2.7.4	HyperTerminal	21
2.7.5	PUTTY	22
2.7.6	Virtual Serial Port Driver	22
CAPITULO 3.....		23
3	Diseño del Proyecto	23
3.1	Diseño General	23
3.1.1	Diseño de la Aplicación	24
3.1.1.1	Especificaciones Funcionales	24
3.1.1.1.1	Declaración de Variables Globales.....	26
3.1.1.1.2	Definición de Constantes.....	27
3.1.1.1.3	Código SmartGPS	28
3.1.1.1.4	Código Memoria MMC/SD	35
3.1.1.1.5	Programa Principal	40
3.1.2	Diseño del Hardware	48
3.1.2.1	Especificaciones Funcionales	48
3.1.2.1.1	El Microcontrolador.....	52
3.1.2.1.2	Tarjeta de Desarrollo SmartGPS	53
3.1.2.1.3	Tarjeta ET-MINI SD/MMC.....	54
3.1.2.1.3.1	Especificaciones	56
3.1.2.1.4	Tarjeta de Memoria MMC	57
3.1.2.1.4.1	Conexiones y Señales	57

3.1.2.1.5	Tarjeta de Memoria SD	59
3.1.2.1.5.1	Conexiones y Señales	60
3.1.2.1.6	MAX232CPE. Transciver RS232	61
3.1.2.1.7	El dsPIC	62
3.1.2.1.8	Sistema de Desarrollo dsPIC PRO4	63
CAPITULO 4	68
4	Simulación y Pruebas Experimentales	68
4.1	Verificación de la Transmisión de Datos del Módulo GPS	68
4.1.1	HyperTerminal	69
4.1.2	Programa PUTTY	70
4.1.3	Terminal Virtual del Proteus	71
4.2	Simulaciones Realizadas	72
4.2.1	Ejemplo MMC_FAT_16 en Proteus	72
4.2.2	Proyecto en Proteus	74
4.2.2.1	Resultados de la Simulación del Proyecto	75
4.3	Implementación del Proyecto	77
4.3.1	Resultados de la Implementación del Proyecto	78
CONCLUSIONES	
RECOMENDACIONES	
ANEXOS	
BIBLIOGRAFIA	

INDICE DE FIGURAS

Figura 2.1 Sistema de Triangulación Satelital.....	8
Figura 2.2 Protocolo NMEA	10
Figura 2.3 Tipos de Comunicación Serial	13
Figura 2.4 Transmisión Serial Asíncrona.....	15
Figura 2.5 Bus de Señales del Protocolo SPI	16
Figura 3.1 Diagrama de Bloques del Proyecto	24
Figura 3.2 Caracteres ASCII.....	30
Figura 3.3 Atributos para la Creación de un Archivo.....	38
Figura 3.4 Tarjeta de Transmisión USART	50
Figura 3.5 Tarjeta Principal.....	51
Figura 3.6 El Microcontrolador	52
Figura 3.7 Tarjeta de Desarrollo SmartGPS	54
Figura 3.8 Tarjeta de Desarrollo ET-MINI SD/MMC.....	55
Figura 3.9 Esquemático de la Tarjeta ET-MINI SD/MMC	56
Figura 3.10 Tarjeta de Memoria MMC	57
Figura 3.11 Tarjeta de Memoria SD.....	59
Figura 3.12 Configuración del MAX232	62
Figura 3.13 El dsPIC.....	63
Figura 3.14 Sistema de Desarrollo dsPIC PRO4	64
Figura 3.15 Selección de Fuente de Poder.....	65
Figura 3.16 Programador USB 2.0.....	65
Figura 3.17 Comunicación RS232	66
Figura 3.18 Tarjeta MMC/SD	67

Figura 4.1 Prueba con el HyperTerminal	69
Figura 4.2 Prueba con el Programa PUTTY	70
Figura 4.3 Prueba realizada con Proteus usando el Virtual Terminal	71
Figura 4.4 Simulación del Ejemplo MMC_FAT_16	73
Figura 4.5 Archivo Grabado en la Simulación.....	74
Figura 4.6 Simulación del Proyecto en Proteus	75
Figura 4.7 Resultados de la Simulación.....	76
Figura 4.8 Implementación Física del Proyecto	77
Figura 4.9 Resultados de la Implementación.....	79

INDICE DE TABLAS

Tabla 1.1 Modelos de Tarjetas GPS	5
Tabla 2.1 Pines del Conector DB9.....	18
Tabla 2.2 Pines Conector DB25.....	18
Tabla 3.1 Pines de la Memoria MMC.....	58
Tabla 3.2 Pines de la Memoria SD	60

INTRODUCCION

En este proyecto, mediante la utilización de un módulo de Sistema de Posicionamiento Global, GPS por sus siglas en inglés, instalado en una tarjeta de desarrollo, se procederá a la obtención de cadenas correspondientes al protocolo NMEA. Información que puede ser procesada, a través de un microcontrolador para posteriormente ser almacenada en una memoria MMC/SD mediante el uso de la tarjeta ET-MINI SD/MMC.

En el Capítulo 1 se describe el sistema en general para conocer su alcance y limitaciones, así como también la estrategia implementada. En este capítulo se analizan también otros productos similares en el mercado y se enumeran un conjunto de aplicaciones para el prototipo.

En el Capítulo 2 se examina la fundamentación teórica referente a cada uno de los dispositivos y herramientas utilizadas en la implementación del sistema.

En el Capítulo 3 se detalla el diseño general, en el cual se establece las características relevantes de cada uno de los dispositivos empleados. Además se expone el código de programación del microcontrolador.

En el Capítulo 4 se presentan las simulaciones realizadas, detalles de la implementación y datos obtenidos en diferentes pruebas, así como los resultados finales del proyecto.

Finalmente se exponen las conclusiones y recomendaciones.

CAPITULO 1

1 Descripción General del Sistema

En este proyecto, se ha utilizado una tarjeta de desarrollo GPS, un microcontrolador, una tarjeta ET-MINI SD/MMC y una memoria MMC/SD. Mediante este sistema se puede almacenar sentencias NMEA en una memoria MMC/SD, para su posterior utilización en mapas de posicionamiento.

1.1 Alcance y Limitaciones del Proyecto

A continuación se analiza el alcance de este proyecto y cuáles fueron sus limitaciones, tanto en la implementación física como a nivel de programación.

1.1.1 Descripción del Proyecto

El propósito de este proyecto es la de implementar un sistema capaz de realizar almacenamiento masivo de sentencias NMEA de un módulo GPS en memorias MMC/SD a través de la programación de un microcontrolador, para luego utilizar estas tramas almacenadas en la determinación de mapas de posicionamiento y trayectoria.

1.1.2 Estrategia Implementada

El módulo utilizado en este proyecto es el GPS LEA-5S de la marca U-Blox. Se trata de un chip receptor completamente autónomo y muy versátil, capaz de combinar una serie de funciones con opciones de conectividad flexibles. Su facilidad de integración resulta muy conveniente para aplicaciones móviles, comerciales e industriales.

La interfaz con el mundo exterior es serial a través del puerto UART del microcontrolador, un PIC18F4520. Mediante este micro se puede comunicar con cualquier otro dispositivo que soporte comunicación serial, ya sea con el formato SPI o con el formato I2C.

Al mismo tiempo, este microcontrolador recibirá la trama de datos provista por el GPS, obtendrá los datos necesarios, y los procesará para que finalmente la información sea guardada a través de la tarjeta ET-MINI SD/MMC en la memoria MMC/SD.

Para poder comunicar el microcontrolador con la tarjeta ET-MINI SD/MMC, la cual brinda un medio físico para interactuar con la memoria MMC/SD; se procedió a utilizar el protocolo de comunicación síncrono SPI, pues estos tipos de memorias para asegurar compatibilidad con la mayoría de controladores existentes ofrecen un modo de comunicación alternativo basado en este estándar.

1.1.3 Limitaciones del Proyecto

Una de las principales limitaciones de este proyecto es la poca capacidad de recepción de señales del módulo GPS dentro de espacios cerrados. No funciona muy bien en instalaciones cuyas paredes sean gruesas o que no tengan ventanas.

Otra limitación es que la tarjeta de desarrollo requiere de una alimentación superior a los 5 voltios de corriente directa y puede soportar hasta 25 voltios de corriente directa, ya que tiene incorporado un convertidor de voltaje DC a DC para llevar el valor de entrada hasta los 3.3 voltios de corriente directa con los que trabaja el chip GPS. Este valor de entrada de la fuente de alimentación se ve reflejado en los pines de voltaje de interacción de la tarjeta de desarrollo.

De esta manera resulta necesaria la implementación de una etapa de regulación de voltaje para alimentar al microcontrolador.

Por otro lado la precisión de este dispositivo es de aproximadamente 5 metros, en menores distancias la lectura no representa variaciones.

Referente al uso de la memoria (MMC o SD), es requisito imprescindible que esta tenga formato FAT16 o en su defecto formato FAT; puesto que solamente en este sistema de archivo se puede inicializar la memoria al utilizar la función *Mmc_Fat_Init*, la cual se encuentra en las librerías del compilador mikroBasic PRO for PIC.

Un gran inconveniente es que la tarjeta ET-MINI SD/MMC trabaja en un rango de alimentación de 2,6 voltios a 3,7 voltios; por lo cual fue necesario colocar un par de diodos que establecieran este rango de voltaje para poder alimentar correctamente la tarjeta. En cuanto a las señales del microcontrolador que se conectan directamente a la ET-MINI SD/MMC y que poseen 5 voltios, se tuvo que reducirlos a 3,3 voltios mediante la utilización de un esquema en base a resistencias.

1.2 Análisis de Soluciones Existentes en el Mercado

Actualmente los GPS están presentes en muchas aplicaciones. Se ha llegado a desarrollar una interfaz tan amigable con los usuarios que es utilizada en actividades deportivas, recreacionales y de seguridad.

Los dispositivos existentes en el mercado son de muy grandes prestaciones, pero han sido concebidos para un propósito desde el inicio y resulta complicado alterar su estructura.

En este proyecto se ha utilizado un módulo GPS montado en una tarjeta de desarrollo, la cual puede interactuar con otros dispositivos, esto da la oportunidad de crear o modificar un controlador a la medida y necesidad del sistema que se implementa en este trabajo.

A continuación se presenta un cuadro comparativo en la Tabla 1.1 con algunos modelos de tarjetas de desarrollo basados en módulos GPS.

Modelo	Fabricante	Costo	Comunicación	Antena Externa
Smart GPS LEA5S	Mikroe	\$59.50	SPI, USB, I2C	sí
V23993-EVA1035	Vincon	\$175.00	USB	sí
GPS Dataloger kit	Parallax	\$149.00	RS232, USB	no

Tabla 1.1 Modelos de Tarjetas GPS

1.3 Aplicaciones

- La tendencia principal de este proyecto es la de proporcionar un sistema que sea capaz de realizar almacenamiento masivo de sentencias NMEA en memorias de tipo MMC o SD; aunque también puede ser adaptado el sistema para almacenar en dispositivos USB. Por lo tanto esta aplicación puede ser utilizada en la posterior visualización de mapas, mediante el uso de los datos guardados a través de la creación de un software especializado para llevar a cabo este objetivo.
- Este proyecto puede ser adaptado para que mediante la utilización de un módulo de Sistema de Posicionamiento Global, sea posible determinar las coordenadas de la ubicación de un robot. Información que puede ser procesada, a través de un microcontrolador y así poder mostrar dicha ubicación mediante una interfaz gráfica.

- El proyecto puede ser aplicado como una herramienta interactiva con los usuarios, mostrándole su posición en el globo. Esta implementación se encontrará incorporada en el sistema móvil. Con esto se aportará una ayuda a sistemas de rastreos, sean estos terrestres, marítimos o aéreos.
- El proyecto puede ser acoplado a tener funciones inalámbricas de tal manera que una vez que el microcontrolador decodifica las tramas, estas sean enviadas por medio de un módulo GSM a cualquier lugar donde se encuentre el receptor del sistema y que disponga de una red GSM o redes compatibles con esta. Esta transmisión es factible con ayuda de los comandos AT, que deben ser utilizados en el microcontrolador. De esta manera se podría monitorear de manera independiente la ubicación del móvil que tenga instalado el dispositivo.

CAPITULO 2

2 Fundamentación Teórica

En este capítulo se pretende abarcar todo fundamento teórico en el que se ha basado este proyecto.

2.1 ¿Qué es un GPS y cómo funciona?

El GPS es un sistema de posicionamiento global que permite determinar en todo el mundo la posición de un objeto, de un animal o inclusive de un ser humano. Se conoce que fue mentalizado por los franceses y belgas, pero fueron los estadounidenses quienes lo desarrollaron y lo implementaron, hoy por hoy es operado por el Departamento de Defensa de los Estados Unidos. Actualmente se cuenta con 27 satélites artificiales, 24 de ellos operativos y 3 de respaldo, con trayectorias sincronizadas para cubrir todo el globo.

Para que un receptor pueda determinar su ubicación requiere de la información de al menos 3 satélites de la red, de los que recibe señales de la ubicación y del reloj de cada uno de ellos. Con base en estas señales, el aparato sincroniza el reloj del GPS y calcula el retraso de las

señales (es decir, la distancia al satélite). Por "triangulación" calcula la posición en que este se encuentra, tal como se muestra en la figura 2.1. En el caso del GPS, la triangulación se basa en determinar la distancia de cada satélite respecto al punto de medición. Conocidas las distancias, se determina fácilmente la propia posición relativa respecto a los tres satélites. Conociendo además las coordenadas o posición de cada uno de ellos por la señal que emiten, se obtiene la posición absoluta o las coordenadas reales del punto de medición. También se consigue una exactitud extrema en el reloj del GPS, similar a la de los relojes atómicos que llevan a bordo cada uno de los satélites.

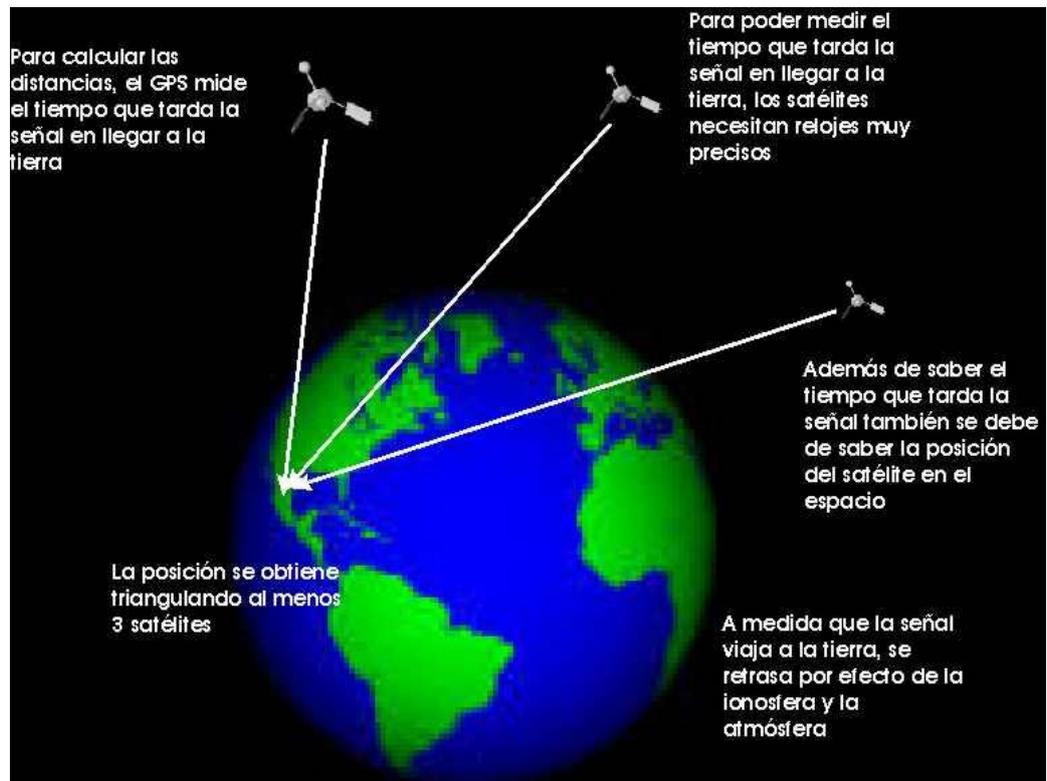


Figura 2.1 Sistema de Triangulación Satelital

2.2 El Protocolo NMEA

NMEA 0183 (o NMEA de forma abreviada) es una especificación combinada eléctrica y de datos entre aparatos electrónicos marinos y, también, en su mayoría, entre receptores GPS.

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

El protocolo NMEA se basa en cadenas. Cada cadena se inicia con el signo \$ (código ASCII 36) y termina con una secuencia de signos que comienza con una nueva línea, tales como CR (código ASCII 13) y LF (código ASCII 10). El significado de toda la cadena depende de la primera palabra. Por ejemplo, una cadena que comienza con \$GPGLL da información acerca de la latitud y la longitud, la hora exacta (Tiempo Universal Coordinado), la validez de los datos (A - Activo o V - Prohibido) y la suma de verificación ("checksum") que permite comprobar si los datos se recibieron correctamente. Los datos individuales de cada elemento están separados por una coma ",".

Cada segundo se envía un conjunto de cadenas NMEA al microcontrolador. En el caso de que los datos sobre latitud y longitud no

sean fijos (por ejemplo, si un receptor GPS falla a la hora de determinar su posición) o cuando los datos no sean identificados, el receptor GPS mantendrá en su salida el mismo juego de cadenas, dejando de lado cualquier dato perdido.

A continuación se puede apreciar la estructura del protocolo NMEA en la figura 2.2

```

gps - HyperTerminal
Archivo Edición Ver Llamar Transferir Ayuda
$GPVTG, .T, .M, 0.161, N, 0.298, K, A=26
$GPGGA, 231035.00, 0208.69068, S, 07958.07253, W, 1.05, 2.57, 78.5, M, 10.8, M, , *68
$GPGSA, A, 3, 19, 32, 03, 20, 06, . . . . ., 5.09, 2.57, 4.40*00
$GPGSV, 3, 1, 11, 03, 66, 156, 35, 06, 64, 115, 21, 11, 08, 229, ., 14, 29, 116, *77
$GPGSV, 3, 2, 11, 16, 39, 004, ., 19, 40, 184, 46, 20, 31, 293, 25, 22, 03, 146, *7D
$GPGSV, 3, 3, 11, 23, 08, 333, ., 31, 11, 044, ., 32, 57, 274, 33*43
$GPGLL, 0208.69068, S, 07958.07253, W, 231035.00, A, A=6B
$GPRMC, 231036.00, A, 0208.69077, S, 07958.07270, W, 0.164, ., 200410, ., A=7A
$GPVTG, .T, .M, 0.164, N, 0.305, K, A=26
$GPGGA, 231036.00, 0208.69077, S, 07958.07270, W, 1.05, 2.57, 78.6, M, 10.8, M, , *67
$GPGSA, A, 3, 19, 32, 03, 20, 06, . . . . ., 5.09, 2.57, 4.40*00
$GPGSV, 3, 1, 11, 03, 66, 156, 35, 06, 64, 115, 22, 11, 08, 229, ., 14, 29, 116, *74
$GPGSV, 3, 2, 11, 16, 39, 004, ., 19, 40, 184, 46, 20, 31, 293, 25, 22, 03, 146, *7D
$GPGSV, 3, 3, 11, 23, 08, 333, ., 31, 11, 044, ., 32, 57, 274, 32*42
$GPGLL, 0208.69077, S, 07958.07270, W, 231036.00, A, A=67
$GPRMC, 231037.00, A, 0208.69085, S, 07958.07289, W, 0.033, ., 200410, ., A=73
$GPVTG, .T, .M, 0.033, N, 0.061, K, A=24
$GPGGA, 231037.00, 0208.69085, S, 07958.07289, W, 1.05, 2.57, 78.7, M, 10.8, M, , *6C
$GPGSA, A, 3, 19, 32, 03, 20, 06, . . . . ., 5.09, 2.57, 4.40*00
$GPGSV, 3, 1, 11, 03, 66, 156, 35, 06, 64, 115, 23, 11, 08, 229, ., 14, 29, 116, *75
$GPGSV, 3, 2, 11, 16, 39, 004, ., 19, 40, 184, 46, 20, 31, 293, 24, 22, 03, 146, *7C
$GPGSV, 3, 3, 11, 23, 08, 333, ., 31, 11, 044, ., 32, 57, 274, 31*41
$GPGLL, 0208.69085, S, 07958.07289, W, 231037.00, A, A=6D
0:00:15 conectado Autodetect. 9600 8-N-1 DESPLAZAR MAY NUM Capturar Imprimir

```

Figura 2.2 Protocolo NMEA

El dispositivo utilizado, LEA-5S, muestra las siguientes cadenas de códigos:

\$GPRMC Recommended minimum specific GPS/Transit data

\$GPVTG Track made good and ground speed

\$GPGGA Global Positioning System Fix Data

\$GPGSA GPS DOP and active satellites

\$GPGSV GPS DOP and active satellites

\$GPGLL Geographic position, latitude / longitude and time

En el caso particular, para la resolución del proyecto nos interesará la trama correspondiente al Sistema Global de Posicionamiento, que es la trama GPGGA, que consta de 15 campos que se separan mediante ‘,’ como se indicó anteriormente. A continuación se muestra un ejemplo de una cadena NMEA completa:

**\$GPGGA,222015.00,0208.66005,S,07958.01067,W,2,10,0.93,101.3,M,10.8,
M,,*5B**

La cadena \$GPGGA provee de datos acerca de la longitud, latitud, hora UTC y otros. La información que nos brindan estos campos es la siguiente:

1. Hora UTC (Tiempo Universal Coordinado) en formato: hhmmss
2. Latitud en formato: ggmm.ssss
3. Orientación en latitud: N (norte) o S (sur)

4. Longitud en formato: gggmm.ssss
5. Orientación en longitud: E (este) o W (oeste)
6. Indicación de calidad GPS: 0=nula; 1=GPS fija; 2=Diferencial GPS
fija
7. Número de satélites visibles por el receptor: nn
8. Dilución horizontal de posición: xx.x
9. Altitud de la antena sobre el nivel del mar (geoidal): xxxxx.x
10. Unidades de altitud: M (metros)
11. Separación geoidal: xxx.x
12. Unidades de separación: M (metros)
13. Tiempo en segundos desde la última actualización: xx
14. ID de referencia de la estación
15. Checksum: *xx

2.3 Modos de Transmisión de Datos

Según el sentido de la transmisión podemos encontrarnos con tres tipos diferentes:

Simplex: Un equipo transmite, el otro recibe.

Half-duplex: Transmiten ambos equipos pero no simultáneamente; los equipos se alternan en la transmisión, uno transmite mientras el otro recibe.

Full-duplex: Ambos equipos transmiten simultáneamente. Para ello se requieren dos líneas independientes, transmisión y recepción; la línea de transmisión de un equipo se conecta a la entrada de recepción del otro y viceversa. Los puertos serie del PC son capaces de utilizar este modo. Véase figura 2.3.

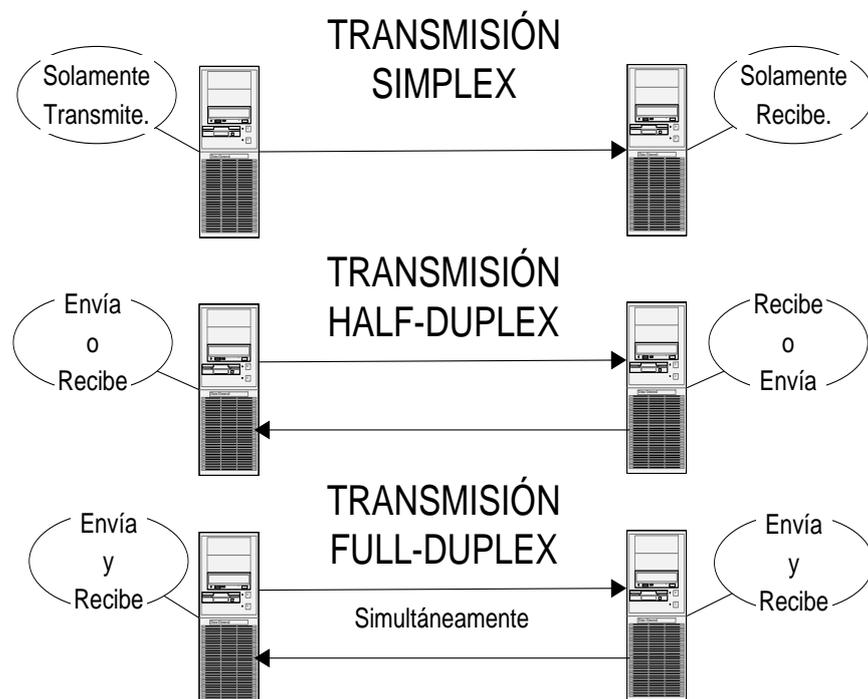


Figura 2.3 Tipos de Comunicación Serial

2.4 Comunicación Serial

Una de las mayores ventajas de la aplicación que se implementó en el prototipo fue la de contar con la comunicación serial.

En la comunicación serie los bits se transmiten uno detrás de otro (de ahí el nombre), lo que hace que sea mucho más lenta que su homóloga "paralelo" en la que se transmiten varios bits a la vez.

La ventaja es que puede utilizarse un solo par de hilos, o incluso uno solo, si el retorno se realiza por la tierra.

2.4.1 Tipos de Comunicación Serial

Existen dos tipos de comunicaciones seriales: la síncrona y asíncrona.

En la comunicación serial síncrona, se necesitan 2 líneas, una línea sobre la cual se transmitirán los datos y la otra contendrá los pulsos de reloj que indicaran cuando un dato es válido. Ejemplo de este tipo de comunicación es el protocolo SPI, el cual se utiliza en el desarrollo del proyecto.

En la comunicación Serial asíncrona, no son necesarios los pulsos de reloj, puesto que tanto el transmisor como el receptor tienen incluido el reloj de sincronización de tal forma que no se transmite a lo largo de la línea de comunicación.

La duración de cada bit está determinada por la velocidad con la cual se realiza la transferencia de datos.

La figura 2.4 muestra la estructura de un carácter que se transmite en forma serial asíncrona.

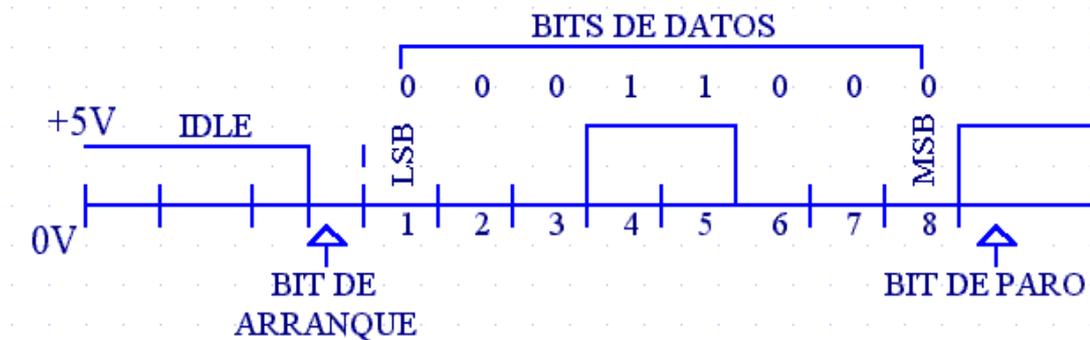


Figura 2.4 Transmisión Serial Asincrónica

2.4.1.1 El Protocolo SPI

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

Algunos dispositivos solo pueden ser transmisores y otros solo receptores, generalmente un dispositivo que tramite datos también puede recibir.

Los dispositivos conectados al bus son definidos como maestros y esclavos. Un maestro es aquel que inicia la transferencia de información sobre el bus y genera las señales de reloj y control. Un esclavo es un dispositivo controlado por el maestro. Cada esclavo es

controlado sobre el bus a través de una línea selectora llamada **Chip Select o Select Slave**, por lo tanto el esclavo es activado solo cuando esta línea es seleccionada. Generalmente una línea de selección es dedicada para cada esclavo.

En un tiempo determinado T1, solo podrá existir un maestro sobre el bus. Cualquier dispositivo esclavo que no esté seleccionado, debe deshabilitarse (ponerlo en alta impedancia) a través de la línea selectora (**chip select**).

El bus SPI emplea un simple registro de desplazamiento para transmitir la información.

Este estándar incluye una línea de reloj SCLK, dato entrante, dato saliente y un pin select SS, que conecta o desconecta la operación del dispositivo con el que uno desea comunicarse. Véase figura 2.5.

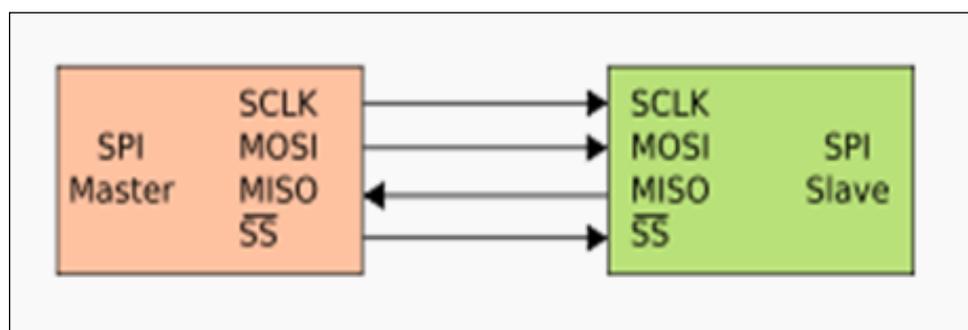


Figura 2.5 Bus de Señales del Protocolo SPI

Las señales que se manejan en el protocolo SPI son:

- SCLK: Clock Serial (Salida para el master)
- MOSI/SIMO: Salida para el master, Entrada para el esclavo

- MISO/SOMI: Entrada para el Master Entrada, Salida para el esclavo
- SS: Chip Select (Se activa en bajo; Salida para el master)

2.4.1.2 La Norma RS-232

Consiste en un conector tipo DB-25 de 25 pines, aunque es normal encontrar la versión de 9 pines DB-9, más barato e incluso más extendido para cierto tipo de periféricos (como el ratón serie del PC). En cualquier caso, los PCs no suelen emplear más de 9 pines en el conector DB-25. Las señales con las que trabaja este puerto serial son digitales, de +12V (0 lógico) y -12V (1 lógico), para la entrada y salida de datos, y a la inversa en las señales de control. El estado de reposo en la entrada y salida de datos es -12V. Dependiendo de la velocidad de transmisión empleada, es posible tener cables de hasta 15 metros. Cada pin puede ser de entrada o de salida, teniendo una función específica cada uno de ellos. Las más importantes se observan en las tablas 2.1 y 2.2.

Las señales TXD, DTR y RTS son de salida, mientras que RXD, DSR y CTS son de entrada. La tierra de referencia para todas las señales es SG (Tierra de Señal). Finalmente, existen otras señales como RI (Timbre Telefónico).

#	Pin	E/S	Función	Conector DB 9
1			Tierra de Chasis	
2	RXD	E	Recibir Datos	
3	TXD	S	Transmitir Datos	
4	DTR	S	Terminal de Datos Listo	
5	SG		Tierra de señal	
6	DSR	E	Equipo de Datos Listo	
7	RTS	S	Solicitud de Envío	
8	CTS	E	Libre para Envío	
9	RI	S	Timbre Telefónico	

Tabla 2.1 Pines del Conector DB9

#	Pin	E/S	Función	Conector DB 25
1			Tierra de Chasis	
2	TXD	S	Transmitir Datos	
3	RXD	E	Recibir Datos	
4	RTS	S	Solicitud de Envío	
5	CTS	E	Libre para Envío	
6	DSR	E	Equipo de Datos Listo	
7	SG		Tierra de señal	
8	CD/DCD	E	Detector de Portadora	
15	TxC	S	Transmitir Reloj	
17	RxC	E	Recibir reloj	
20	DTR	S	Terminal de Datos Listo	
22	RI	S	Timbre Telefónico	
24	RTxC	S/E	Transmitir/Recibir Reloj	

Tabla 2.2 Pines Conector DB25

2.5 Sistema de Archivos

No existe un sistema de archivos suficientemente desarrollado para tarjetas de memoria Flash SD. La gran mayoría de dispositivos electrónicos con memorias Flash utilizan el sistema de archivos FAT, de Microsoft, más concretamente en la versión FAT16. Esto es debido a que normalmente, cualquiera de esos dispositivos necesita de comunicación con un ordenador personal, y a día de hoy la gran mayoría de ordenadores funcionan con sistemas operativos de Microsoft.

2.6 Conceptos Básicos de Programación

En esta sección se realizará una breve descripción de los conceptos de programación usados en el presente trabajo.

Variables Globales. Una variable global es en informática, una variable accesible en todos los ámbitos de un programa. Los mecanismos de interacción con variables globales se denominan mecanismos de entorno global. El concepto de entorno global contrasta con el de entorno local donde todas las variables son locales sin memoria compartida.

Variables Locales. Es la variable a la que se le otorga un ámbito local. Tales variables sólo pueden accederse desde la función o bloque de instrucciones en donde se declaran. Las variables locales se

contraponen a las variables globales. En la mayoría de lenguajes de programación las variables locales son variables automáticas almacenadas directamente en la pila de llamadas. Esto significa que cuando una función recursiva se llama a sí misma, las variables locales reciben, en cada instancia de la función, espacio para el direccionamiento de memoria separado. De esta forma las variables con este ámbito se pueden declarar, reescribir y leer sin riesgo de efectos secundarios para los procesos fuera del bloque en el que son declarados.

Funciones. Llamadas también subprogramas o subrutinas, se presenta como un sub-algoritmo que forma parte del algoritmo principal, el cual permite resolver una tarea específica y devuelve un valor.

Procedimientos. Los procedimientos se asemejan mucho a las funciones, con la única diferencia que no devuelven un valor. Son utilizados para el procesamiento de información sobre las variables de ámbito global.

2.7 Herramientas de Software

En esta sección se presentan las herramientas de software utilizadas tanto para la simulación como para la implementación del proyecto.

2.7.1 MikroBasic PRO for PIC.

Es un compilador avanzado para microcontroladores PIC. Su plataforma de programación es en Basic. La versión Pro incluye un conjunto de librerías y ejemplos destinados a facilitar el desarrollo de aplicaciones.

2.7.2 MikroC PRO for dsPIC.

Es un compilador avanzado para microcontroladores dsPIC. Su plataforma de programación es en lenguaje C. Esta versión incluye un conjunto de librerías y ejemplos destinados a facilitar el desarrollo de aplicaciones para DSPIC PRO4.

2.7.3 Proteus

Es un programa para simular circuitos electrónicos complejos integrando inclusive desarrollos realizados con microcontroladores de varios tipos, en una herramienta de alto desempeño con capacidades graficas. La versión utilizada fue 7.6.4.

2.7.4 HyperTerminal

HyperTerminal es un programa que se puede utilizar para conectar con otros equipos, sitios Telnet, servicios en línea y equipos host, mediante un módem, un cable de módem nulo o una conexión con protocolo Internet. Permite además visualizar los datos en formato de texto que recibe un computador por su puerto serial.

2.7.5 PUTTY

A partir de la versión de Windows Vista, ya no se encuentra disponible el programa Hyperterminal. El programa PUTTY es un cliente Telnet con licencia libre. Este programa nos permite emular una conexión del puerto serial y visualizar los datos recibidos.

2.7.6 Virtual Serial Port Driver

Hoy en día, un computador personal o portátil, no dispone ya de un puerto RS232, únicamente puertos USB. Mediante una aplicación para controlar puertos virtuales como el Virtual Serial Port Driver es posible configurar uno de esos puertos USB como puerto RS232.

CAPITULO 3

3 Diseño del Proyecto

En el presente capítulo se ponen de manifiesto todas las etapas de diseño, implementación y pruebas necesarias para la elaboración de este informe.

3.1 Diseño General

El propósito del proyecto radica en la implementación de un prototipo capaz de comunicar un módulo GPS con un microcontrolador de la familia avanzada, el cual decodificará el protocolo NMEA transferido desde el módulo antes mencionado y seleccionará una trama específica, esta será almacenada continuamente en una memoria de de tipo MMC o SD.

El módulo GPS instalado en la tarjeta de desarrollo, se comunica a través de su puerto de transmisión de datos Tx hacia el microcontrolador.

Mediante un solo hilo de datos, más la línea de tierra, el módulo GPS envía cada segundo un paquete de cadenas NMEA. Estos caracteres

son separados por el microcontrolador seleccionando únicamente la trama de datos que lleve el encabezado \$GPGGA. Luego estos datos son enviados a la tarjeta ET-MINI SD/MMC por medio de la comunicación serial síncrona, utilizando el protocolo SPI. Mediante esta tarjeta de desarrollo se almacenaran los datos procesados en una memoria MMC/SD. Véase figura 3.1.

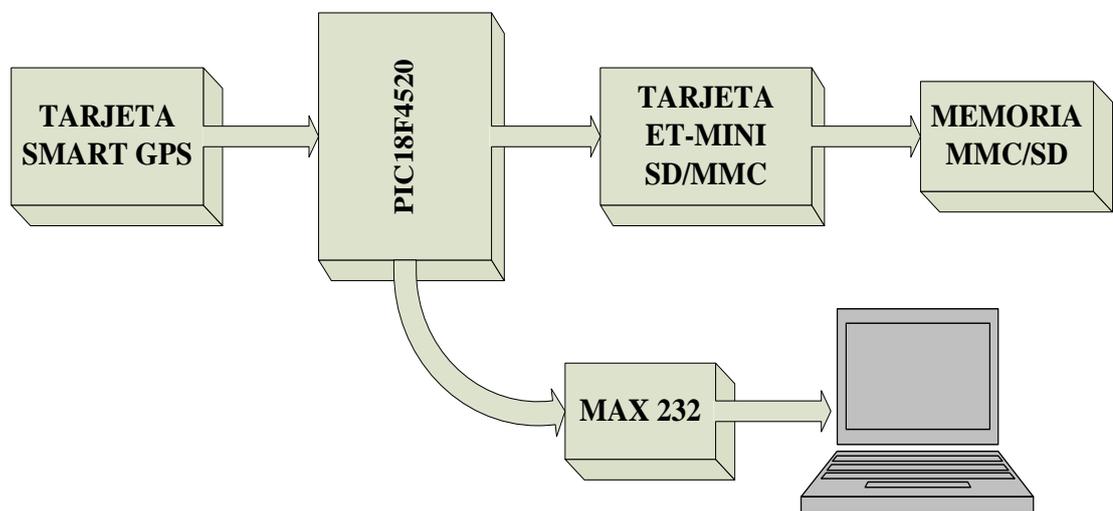


Figura 3.1 Diagrama de Bloques del Proyecto

3.1.1 Diseño de la Aplicación

En esta sección se analiza cada uno de los componentes de software que forman parte de este proyecto.

3.1.1.1 Especificaciones Funcionales

Para programar el PIC18F4520 se usa la herramienta de software **MikroBasic PRO for PIC** de Mikroelectrónica, la cual fue descrita en el capítulo 2 del presente trabajo.

El código de programación ha sido particionado en seis bloques principales, para poder sustentar detalladamente las funcionalidades de cada una de estas secciones.

El primer bloque corresponde a la declaración de variables globales que se utilizan a lo largo del programa.

El segundo bloque está dedicado a la definición de constantes que son requeridas en el procedimiento relativo al manejo del módulo GPS.

El tercer bloque se enfoca en lo correspondiente a la tarjeta de desarrollo SmartGPS, en lo referente a la interpretación de las tramas y procesamiento adecuado de estas.

El cuarto bloque está compuesto de tres procedimientos; los cuales se encargan del control y funcionamiento para el adecuado uso de la memoria.

El quinto bloque está dedicado al desglose del programa principal, cuya importancia radica en lograr un correcto funcionamiento de todo

el sistema en conjunto. En este bloque las funciones y procedimientos desarrollados anteriormente pueden ser llamados o invocados varias veces según lo requiera la aplicación cuando está en ejecución.

3.1.1.1.1 Declaración de Variables Globales

A continuación se procede a especificar de manera detallada el bloque de declaración de variables:

```
programCodigoMMC

dim fat_txt as STRING[20]          "FAT16 not found",

dim filename as STRING[14]        "DATOSGPSTXT"

dim caracter,viajero,flag,x,dato_recibido as byte

dim dato as char[80]

dim received_byte as char
```

En primer lugar se encuentra la variable **fat_txt**, que está declarada como una cadena de caracteres de 20 posiciones; esta variable global es utilizada posteriormente en el bloque principal para indicar cuando la tarjeta MMC/SD no ha podido ser inicializada correctamente.

La variable **filename** se encuentra declarada como una cadena de caracteres de 14 posiciones; a esta variable se le asignará el nombre del archivo que se procederá a crear en el momento que se ejecute la aplicación.

Las variables **caracter**, **viajero**, **flag**, **x**, **dato_recibido** son declaradas como byte y son utilizadas en el procedimiento interrupt, que corresponde al procesamiento de las tramas del protocolo NMEA provista por el modulo GPS.

La variable **dato** es declarada como un arreglo de 80 caracteres y en esta se almacenará la cadena seleccionada del protocolo NMEA.

La variable **received_byte** es declarada como carácter y es utilizada en la determinación de la cadena GPGGA.

3.1.1.1.2 Definición de Constantes

En este bloque se presentan las constantes que fueron definidas en el programa:

```
const INICIO = $24
```

```
const FIN = $0A
```

La constante **INICIO** es definida para reemplazar el código ASCII del carácter \$, el cual corresponde a 24. De manera similar, la constante **FIN** es definida en lugar del código ASCII correspondiente al fin de línea, que es representada por 0A.

3.1.1.1.3 Código SmartGPS

En este programa se creó el procedimiento **interrupt**, cuyo objetivo es el generar una interrupción cada vez que reciba vía USART una de las cadenas que forman parte del protocolo NMEA. Este procedimiento además es el encargado de seleccionar la trama a la que le pertenecen las siglas GPGGA.

En este procedimiento se emplea la función **Usart_Data_Ready**, la cual es una función propia de las librerías que proporciona Mikroelectrónica y que es equivalente a escribir en el segmento de código PIR1.RCIF.

Esta función me indica si los datos en el buffer de recepción están listos para ser leídos; por lo tanto me retorna 1 si hay dato en el buffer, caso contrario me retorna cero. Para usar esta rutina es requerido que el módulo USART sea inicializado.

También se utiliza la función **Usart_Read**, propia de las librerías de Microelectrónica; básicamente esta función recibe un byte vía USART y luego retorna dicho byte. De igual manera a la anterior

función, esta rutina requiere que el módulo USART sea inicializado. Una vez que el dato está listo para ser leído y es capturado con la ayuda de las funciones anteriormente explicadas; este dato es almacenado en la variable `received_byte`. A continuación va a ir siendo comparado con el ASCII correspondiente a los caracteres \$ (definido como INICIO), G, P, G, G, A. Si el ASCII es igual al byte recibido; este será almacenado en la variable `dato` en la respectiva posición, sino se repetirá el proceso empezando en el caso 0.

En el momento que un byte recibido tiene coincidencia con el ASCII, el programa avanzará al siguiente caso, incrementando automáticamente el puntero en la variable `dato` y esperando por un nuevo byte listo para ser leído en el buffer de recepción.

Este proceso se repetirá hasta haber determinado la trama \$GPGGA, en cuyo caso se llenará la variable `dato` con todos los caracteres de la cadena ha sido receptada hasta encontrar el ASCII del salto de línea, que está representado por la constante `FIN`; en este instante se iguala la variable `dato_recibido` a uno, indicando que se ha receptado completamente una trama.

Finalmente una vez que se ha obtenido la trama se debe encerrar la bandera de recepción del USART y habilitar nuevamente la

interrupción para que el proceso se siga repitiendo indefinidamente mientras se esté ejecutando la aplicación

La tabla de los caracteres ASCII es mostrada en la figura 3.2.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SPC	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL
8	€	□	,	f	„	...	†	•	—	™	š	‹	œ	□	ž	□
9	□	'	'	"	"	•	—	—	~	™	š	›	œ	□	ž	Ÿ
A		i	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯
B	°	±	²	³	´	µ	¶	·	,	¹	º	»	¼	½	¾	¿
C	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Figura 3.2 Caracteres ASCII

```
sub procedure interrupt()  
  
  if Usart_Data_Ready = 1 then  
  
    received_byte = Usart_Read  
  
    select case viajero  
  
      case 0  
  
        if (received_byte = INICIO) then '$'  
  
          dato[x]= received_byte  
  
          inc(x)  
  
          viajero=1  
  
        else  
  
          viajero=0  
  
        end if  
  
      case 1  
  
        if (received_byte=71) then 'G'  
  
          dato[x]= received_byte  
  
          inc(x)
```

```
        viajero=2

    else

        viajero=0

        x=0

    end if

case 2

    if (received_byte=80) then 'P'

        dato[x]= received_byte

        inc(x)

        viajero=3

    else

        viajero=0

        x=0

    end if

case 3

    if (received_byte=71) then 'G'
```

```
    dato[x]= received_byte

    inc(x)

    viajero=4

else

    viajero=0

    x=0

end if

case 4

    if (received_byte=71) then 'G'

        dato[x]= received_byte

        inc(x)

        viajero=5

    else

        viajero=0

        x=0

    end if
```

case 5

if (received_byte=65) then 'A'

dato[x]= received_byte

inc(x)

viajero=6

else

viajero=0

x=0

end if

case 6

if(received_byte= FIN)then

viajero=7

dato_recibido=1

dato[x]=0

else

dato[x]= received_byte

```
        inc(x)

    end if

case 7

    if(dato_recibido=0)then

        viajero=0

    end if

end select

end if

PIR1.RCIF=0

PIE1.RCIE=1

end sub
```

3.1.1.1.4 Código Memoria MMC/SD

Este bloque de programa esta compuesto por tres procedimientos que se detallan a continuación:

El primer procedimiento es denominado **i_Write_Str**, cuya función radica en escribir cadenas de caracteres por medio del USART, para ser leídas en programas como el HyperTreminal o PUTTY.

Cuando es invocado este procedimiento desde el programa principal, se le envía como parámetro una cadena de caracteres, que esta rutina se encarga de transmitir byte a byte por medio de la función **USART_Write**.

La función USART_Write no retorna nada, pero requiere de la inicialización del módulo USART para poder ser empleada.

```
sub procedure i_Write_Str(dim byref ostr as byte[1])

    dim i as byte

    i = 0

    while (ostr[i] <> 0)

        USART_Write(ostr[i])

        Inc(i)

    wend

    USART_Write(0x0A)

end sub
```

En esta sección se utilizó el procedimiento **M_Create_New_File**, el cual sirve para crear un nuevo archivo y escribir algunos datos sobre este.

Este procedimiento cuenta con las rutinas **Mmc_Fat_Assign** y **Mmc_Fat_Rewrite**, las mismas que pertenecen a las librerías de Microelectrónica.

La función **Mmc_Fat_Assign** retorna uno si el archivo ya existe o si el archivo no existe pero uno nuevo es creado, y retorna cero en caso contrario. Además requiere que tanto la librería MMC como la tarjeta MMC/SD sean correctamente inicializadas.

Esta rutina recibe como parámetros una cadena de caracteres que se empleará para nombrar el archivo; cabe recalcar que el nombre debe estar en formato (nombre del archivo.extensión), por lo que el programa automáticamente considerara los tres últimos caracteres del nombre como la extensión del archivo. También se le envía como parámetro a esta rutina un valor en hexadecimal, en el cual cada bit corresponde al atributo específico que se le designa al archivo en el momento de su creación.

En el caso propio del proyecto se le envía A0 para la creación de un nuevo archivo. En la figura 3.3 se ponen en manifiesto todos los atributos con los cuales se puede crear un archivo.

Bit	Mask	Description
0	0x01	Read Only
1	0x02	Hidden
2	0x04	System
3	0x08	Volume Label
4	0x10	Subdirectory
5	0x20	Archive
6	0x40	Device (internal use only, never found on disk)
7	0x80	File creation flag. If file does not exist and this flag is set, a new file with specified name will be created.

Figura 3.3 Atributos para la Creación de un Archivo

La función `Mmc_Fat_Rewrite` abre el archivo que ha sido asignado para escritura. Si el archivo creado no tiene información, empieza a escribir nuevos datos; pero si este no está vacío el contenido es borrado antes de empezar con la escritura.

Esta rutina no retorna nada y necesita que la librería MMC y la tarjeta MMC/SD estén inicializadas adecuadamente.

```
'sub procedure M_Create_New_File()  
  
    Mmc_Fat_Assign(filename, 0xA0)  
  
    Mmc_Fat_Rewrite()  
  
end sub
```

Finalmente en este bloque se creó el procedimiento **M_Write_File**, cuyo objetivo es el de escribir en el archivo creado después de la última línea que ha sido escrita.

Esta formado por las funciones **Mmc_Fat_Append** y **Mmc_Fat_Write**.

La rutina **Mmc_Fat_Append** primeramente abre el archivo, luego coloca el puntero de archivo después del último byte escrito, para posteriormente continuar escribiendo datos a partir de esta posición.

Mmc_Fat_Write es una rutina que no retorna nada y se encarga de escribir una cierta cantidad de bytes en un archivo abierto que ha sido asignado para escritura. Recibe como parámetro el dato y el número de bytes ha ser escrito.

```
sub procedure M_Write_File()  
  
    Mmc_Fat_Append()  
  
    Mmc_Fat_Write(dato, 80)  
  
end sub
```

3.1.1.1.5 Programa Principal

El programa principal es el procedimiento encargado de lograr un correcto funcionamiento de todo el sistema en conjunto, tal como se refirió anteriormente. Este bloque está estructurado por varias etapas esenciales; que se encargan de acondicionar las herramientas de hardware, de inicializar los módulos que se emplean en el desarrollo del proyecto, además del llamado de procedimientos y funciones según lo requiera la aplicación.

Este bloque en un principio corresponde a la inicialización de variables. A continuación es usado para la preparación de los puertos, en el cual se setea todas las patitas del microcontrolador para señales digitales, se colocan como entradas o salidas los pines según correspondan y se habilitan los registros necesarios para el uso de interrupciones.

Una vez configurados los puertos se procede al respectivo seteo del módulo USART para la lectura de los datos. Es así que

mediante la función **USART_Init** se inicializa el módulo USART con la velocidad deseada; en el caso particular del proyecto la velocidad de transmisión es de 4800 bps. También es empleada la función **Usart_Write_Text**, la cual no retorna nada y tiene como parámetro el texto que se transmitirá vía USART.

Además se utiliza una función denominada **Delay_ms**; esta crea un retraso de software en la duración de milisegundos, tiempo que se emplea como una constante que es enviada como parámetro. El rango de constantes aplicable a esta función depende de la frecuencia del oscilador.

Luego se procede a inicializar la librería FAT, para lo que se usa la función **Spi_Init_Advanced**, imprescindible para la configuración e inicialización del estándar SPI. Esta función es llamada por primera vez con el parámetro MASTER_OSC_DIV64, que determina un rango necesario para la correcta inicialización de la tarjeta MMC/SD entre 300 y 400 KHz; los parámetros DATA_SAMPLE_MIDDLE, CLK_IDLE_LOW, LOW_2_HIGH que también corresponden a la función Spi_Init_Advanced son necesarios para determinar la forma de escritura de la memoria. Además se emplea un retardo de 500 ms para asegurar que la tarjeta inicialice adecuadamente, a través de la función Delay_ms.

Luego se emplea la función **Mmc_Fat_Init** para determinar el estado de la memoria, puesto que esta retorna 0 si la memoria MMC/SD fue detectada e inicializada exitosamente; 1 si el sector fat16 de la memoria MMC/SD no ha sido encontrado y 255 si la memoria MMC/SD no fue detectada.

En caso de que la función retorne 0 es llamada por segunda vez la rutina `Spi_Init_Advanced`, pero con la diferencia de que en esta ocasión tendrá como parámetro `MASTER_OSC_DIV4`, para que el sistema emplee en la grabación de datos un reloj igual a la cuarta parte del oscilador externo colocado al microcontrolador que equivale a un cristal de cuarzo de 20 MHz.

Se coloca nuevamente un retardo de 500 ms y a continuación se hace la llamada al procedimiento `M_Create_New_File`, que fue explicado en la sección 3.1.1.1.4 y básicamente sirve para crear un nuevo archivo y escribir algunos datos sobre este. Además se procede con la rutina `Usart_Write_Text`, en la que se transmitirá vía USART un mensaje de inicio indicando que la memoria esta lista para ser grabada.

En caso de que la función `Mmc_Fat_Init` retorne 1 o 255 se empleará el procedimiento `I_Write_Str`, que fue explicado en la sección 3.1.1.1.4 y cuya función radica en escribir cadenas de

caracteres por medio del USART. Este procedimiento será utilizado con la finalidad de enviar un mensaje indicando que la memoria no puede grabar datos.

Finalmente se hace uso de un lazo infinito en el que se llama al procedimiento `M_Write_File` cada vez que se ha recibido una trama; esto sucede cuando la variable `dato_recibido` es igual a uno. La rutina `M_Write_File` fue detallada en la sección 3.1.1.1.4 y su objetivo conlleva a la escritura en un archivo existente después de la última línea que se ha escrito. Por consiguiente, este lazo infinito ayuda al almacenamiento continuo de las cadenas una a continuación de la otra.

Después de la ejecución de estas rutinas se enceran las variables, preparándolas para la siguiente ejecución en la adquisición de los datos del módulo GPS. Por último se transmite un mensaje vía USART indicando que ha finalizado la escritura de una cadena completa con la función `Usart_Write_Text`.

Las líneas de código correspondientes al programa principal son detalladas a continuación:

main:

'--- Declaración de variables

fat_txt = "FAT16 not found"

filename = "DATOSGPSTXT"

dato_recibido=0

viajero=0

x=0

flag=0

'--- Preparación de los puertos para la señalización

ADCON1 = 0x3F 'Setea todo como señales digitales

TRISA = 0

TRISB = 0

TRISC=%10010000

TRISD = 0

INTCON = %11000000

INTCON2 = %10000000

```
PIE1=%00100000
```

```
PIR1=0
```

```
IPR1=%00100000
```

```
PORTB = 0
```

```
'--- Setear el módulo USART
```

```
USART_Init(4800)
```

```
Delay_ms(300)
```

```
Usart_Write_Text("INICIO PIC:")
```

```
Delay_ms(200)
```

```
'--- Inicializar la librería FAT
```

```
Spi_Init_Advanced(MASTER_OSC_DIV64,  
DATA_SAMPLE_MIDDLE, CLK_IDLE_LOW, LOW_2_HIGH)
```

```
Delay_ms(500)
```

```
if Mmc_Fat_Init(PORTC, 2) = 0 then
```

```
PORTB.0 = 1
```

```
Spi_Init_Advanced(MASTER_OSC_DIV4,  
DATA_SAMPLE_MIDDLE, CLK_IDLE_LOW, LOW_2_HIGH)  
  
Delay_ms(500)  
  
M_Create_New_File()  
  
Usart_Write_Text(":Start:")  
  
else  
  
I_Write_Str(fat_txt)  
  
end if  
  
'--- Almacenamiento de datos  
  
while(1)  
  
if (dato_recibido =1) then  
  
M_Write_File()  
  
dato_recibido=0  
  
x=0  
  
if(flag=0)then  
  
PORTB.0=0
```

```
        flag=1

    else

        PORTB.0=1

        flag=0

    end if

    Usart_Write_Text(":end:")

end if

wend

end.
```

3.1.2 Diseño del Hardware

En esta sección se analiza en detalle cada uno de los componentes del hardware que se utilizaron para llevar a cabo el desarrollo del proyecto.

3.1.2.1 Especificaciones Funcionales

El prototipo está compuesto por 4 tarjetas independientes que se comunican a la placa principal a través de buses de datos.

En primer lugar se presenta una tarjeta de desarrollo denominada SmartGPS, en la cual se encuentra ubicado el módulo GPS LEA-5S de la marca U-Blox de Microelectrónica. La tarjeta de desarrollo se comunica a través de su puerto de transmisión de datos Tx hacia la tarjeta principal. Es así que mediante un bus de datos, en el que consta un solo hilo de datos, más el de alimentación y la línea de tierra se comunica a la tarjeta que contiene el microcontrolador; en donde las señales de alimentación y tierra van directo al integrado 7805, regulando a 5 voltios la alimentación que recibirá todo el sistema.

La tarjeta de desarrollo es alimentada por una fuente externa de 9 V, y es necesario activar la antena externa para poder obtener las tramas que son enviadas del módulo GPS. De la misma manera en el banco de switch de la tarjeta SmartGPS se necesita establecer el

tipo de microcontrolador que se utiliza, en el caso del proyecto es un pic.

Por medio del software que tiene por nombre la misma marca del GPS (U-Blox) se configura el módulo para que la tasa de transmisión sea equivalente a 4800 bps.

Para realizar las pruebas correspondientes a la grabación de la memoria se utilizó en un principio la tarjeta de entrenamiento dsPICPRO4, la cual es descrita en detalle en la sección 3.1.2.1.8.

Una vez entendido el funcionamiento de la memoria y los requisitos necesarios para su grabación se procedió a interactuar con la tarjeta ET-MINI SD/MMC, que sirve de interfaz por medio de la cual se puede grabar la memoria MMC/SD. Esta tarjeta se comunica con el microcontrolador a través de un bus de datos que provee la alimentación y las respectivas señales para poder grabar adecuadamente la memoria.

Se diseñó una placa para la transmisión vía USART, la cual es encargada únicamente de facilitar comunicación serial a una PC en caso de que sea requerido. Este se comunica a la placa principal igualmente a través de un bus de datos. En la figura 3.4 se muestra el diseño de esta tarjeta.



Figura 3.4 Tarjeta de Transmisión USART

El esquemático de la placa principal fue diseñado en el software Altium. Esta tarjeta se encuentra conformada por los siguientes elementos: el PIC18F4520, una botonera empleada como reset del sistema, un led de color rojo; cuyo parpadeo indica la grabación de una cadena completa en la memoria MMC/SD. También se encuentra un cristal de cuarzo de 20MHz; que actúa como oscilador externo para el microcontrolador y un par de capacitores cerámicos; que sirven para completar la configuración del reloj externo.

Esta placa cuenta con el integrado 7805; el cual actúa como regulador de voltaje, un conjunto de resistencias necesarias para

conseguir los valores del rango de voltaje en el cual trabaja la tarjeta ET-MINI SD/MMC. Además, son empleados en la tarjeta tres diodos; los cuales se tienen por función conseguir un voltaje de 3,3 V, que es el necesario para alimentar tarjeta ET-MINI SD/MMC.

Finalmente se utilizaron dos capacitores para mantener el voltaje de alimentación estable. En la figura 3.5 se muestra el diseño e implementación de la tarjeta.

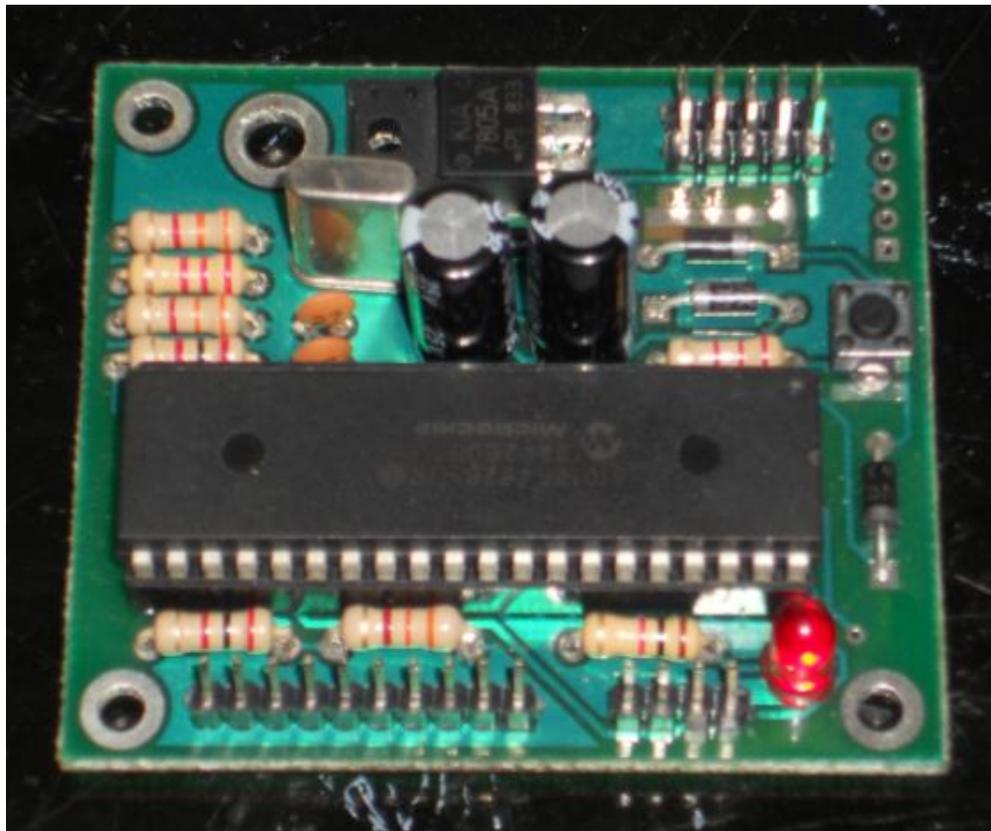


Figura 3.5 Tarjeta Principal

3.1.2.1.1 El Microcontrolador

El PIC18F4520-I/P es un microcontrolador de 8 bits del tipo flash, con memoria de programa de 32KB y una RAM de 1536 Bytes. Posee dos módulos PWM, oscilador interno de hasta 8MHz y convertidor A/D de 13 canales. Además tiene incorporado 4 timers, de los cuales uno es de 8 bits y los otros son de 16 bits. De las 40 patitas que presenta el empaquetado, 36 pueden ser usadas como pines de entradas o salidas.

Tiene un puerto EUSART mejorado, el cual permite comunicación serial compatible con RS232 y RS485.

Este micro posee además un puerto maestro de comunicación serial capaz de manejar formatos SPI (Serial Peripheral Interface) y también I2C (Integer-Integrated Circuit), permitiendo así la comunicación con otros dispositivos compatibles. Véase figura 3.6.

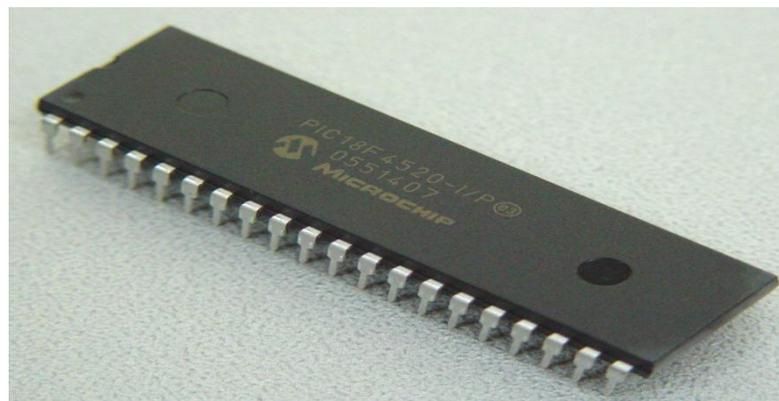


Figura 3.6 El Microcontrolador

3.1.2.1.2 Tarjeta de Desarrollo SmartGPS

El fabricante de esta tarjeta de desarrollo ha utilizado un chip GPS U-BLOX LEA 5S. Tiene incorporada una pila de 3V para respaldo de su información. Cuenta con una etapa de regulación de voltaje mediante un convertidor DC a DC que permite llevar cualquier valor de voltaje hasta 25Vdc en la entrada, a los 3,3V con los que opera el chip GPS.

Soporta los siguientes protocolos de comunicación: UART, USB, y es compatible también con I2C.

Sus pines de comunicación de 3,3V son convertidos a 5V mediante una etapa de acoplamiento con 3 transistores conectados en configuración buffer.

Mediante un banco de interruptores incorporado se puede realizar la selección de comunicación para los siguientes microcontroladores: PICs, dsPICs, AVR y ARM.

Puede recibir además una alimentación externa proveniente de una tarjeta madre compatible con sus funciones. Véase figura 3.7.

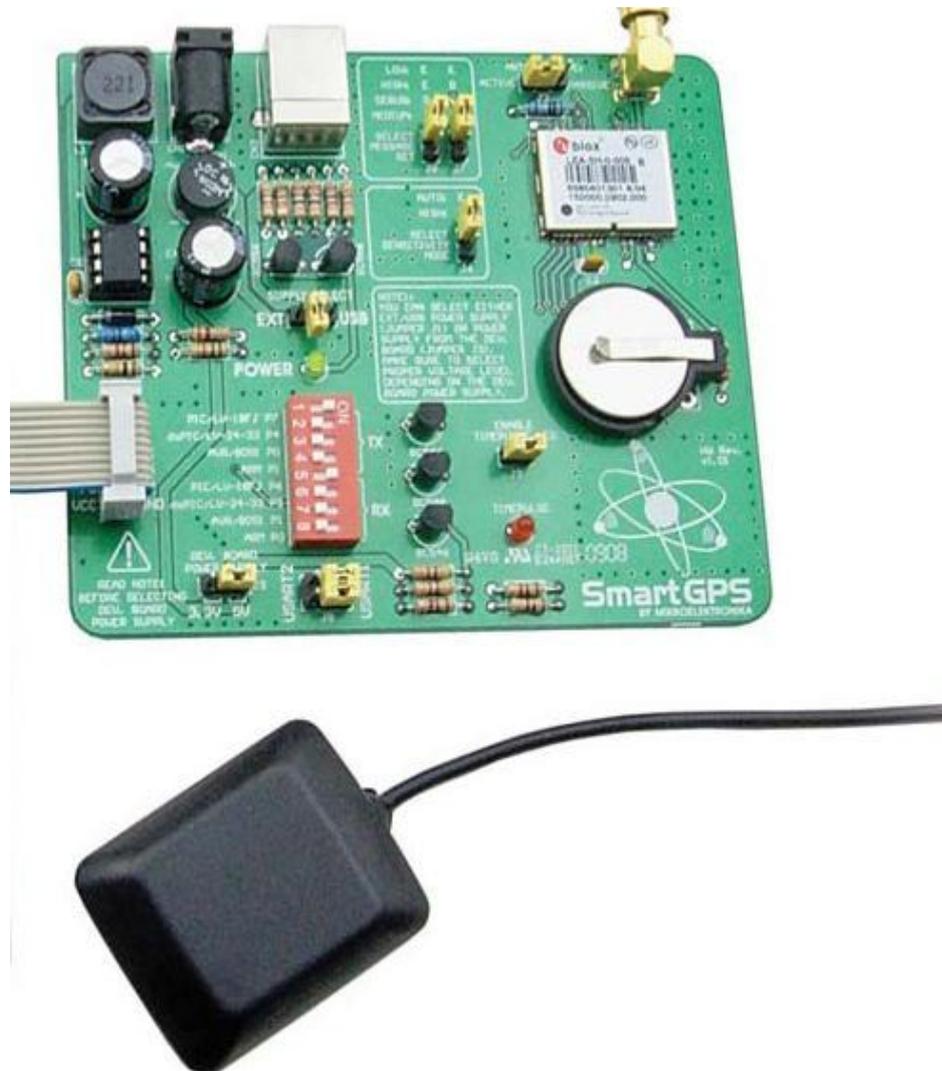


Figura 3.7 Tarjeta de Desarrollo SmartGPS

3.1.2.1.3 Tarjeta ET-MINI SD/MMC

Es una tarjeta de desarrollo que sirve para conectar componentes con memorias SD/MMC; por ejemplo, para el desarrollo del proyecto se utilizó este dispositivo para realizar la conexión entre un microcontrolador y una tarjeta de memoria (SD/MMC).

Este dispositivo contiene el zócalo para insertar la memoria, ya sea de tipo SD o MMC; también provee un conector las señales externas de los pines de la memoria, por lo que resulta muy conveniente para diversas aplicaciones.

Por otra parte, contiene otros circuitos tales como: El circuito Detector de la Memoria, El Circuito de Resistencias Pull-up y otros tal como se muestra en la figura 3.8 y en la figura 3.9.

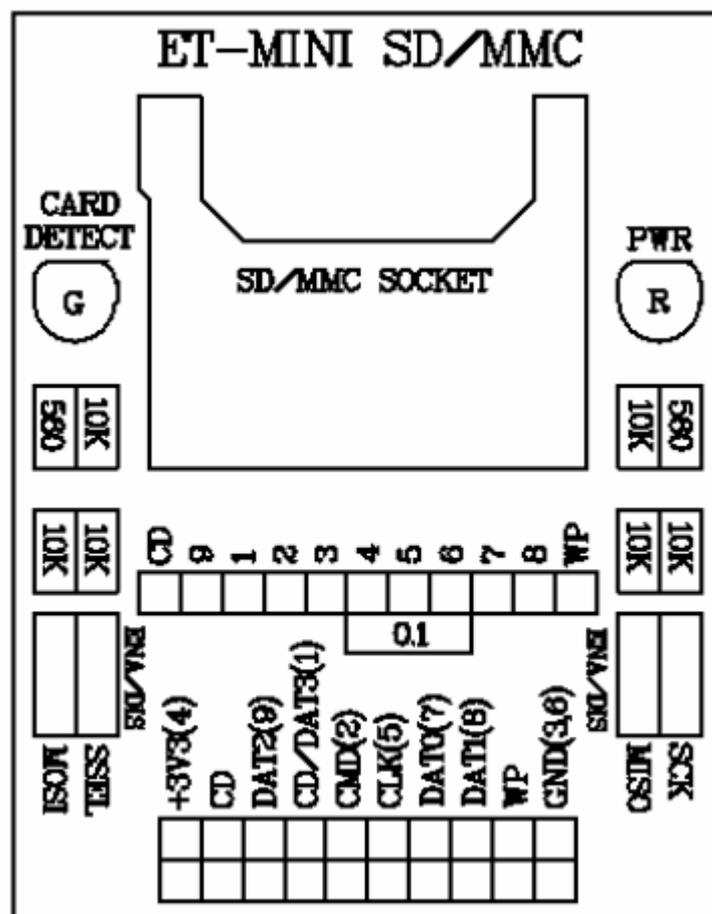


Figura 3.8 Tarjeta de Desarrollo ET-MINI SD/MMC

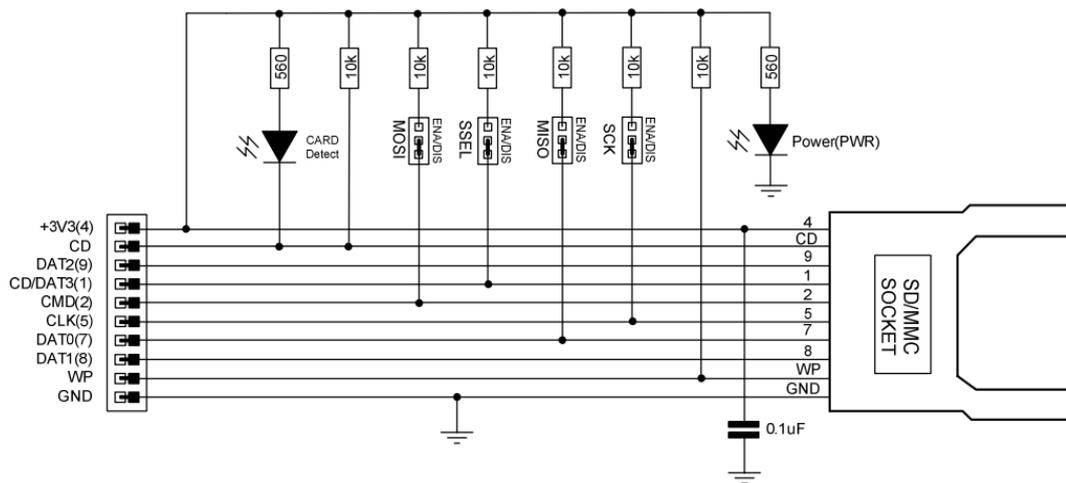


Figura 3.9 Esquemático de la Tarjeta ET-MINI SD/MMC

3.1.2.1.3.1 Especificaciones

- Soporta tarjetas de memorias tipo SD o tipo MMC.
- Se puede activar o desactivar el circuito de las resistencias Pull-up.
- Puede mostrar el estado al momento de la inserción de la memoria por medio del circuito detector de tarjeta mediante el uso del led y envía las señales de salida a través del pin CD.
 - * CD = 1; Esto significa que no se ha detectado la memoria.
 - * CD = 0; Esto significa que se ha detectado la memoria.
- Puede mostrar el estado del switch de Protección de Escritura en la memoria SD/MMC y envía las señales de salida a través del pin WP.

- * WP = 1; Esto significa que el interruptor de Protección contra Escritura está en posición OFF.
- * WP = 0; Esto significa que el interruptor de Protección contra Escritura está en posición ON.

3.1.2.1.4 Tarjeta de Memoria MMC

Las memorias MultiMediaCard son dispositivos de almacenamiento Flash muy versátiles debido a su gran capacidad, y reducido tamaño y precio. Véase figura 3.10.



Figura 3.10 Tarjeta de Memoria MMC

3.1.2.1.4.1 Conexiones y Señales

El protocolo de acceso SPI se puede implementar mediante cualquier controlador ordenador personal ó microcontrolador; que disponga de un canal SPI hardware o emulado por software. Cuando se opera en este modo los pines de la memoria MMC se denominan del siguiente modo:

# Pin:	Nombre:	Tipo:	Descripción SPI:
1	!CS	Entrada	Chip Select (activo a 0)
2	DataIn	Entrada	Línea de datos y comandos hacia la memoria.
3	VSS1	Alimentación	Suministro de voltaje de tierra
4	VDD	Alimentación	Alimentación
5	CLK	Entrada	Clock
6	VSS2	Alimentación	Suministro de voltaje de tierra
7	DataOut	Salida	Línea de datos y status de la memoria al microcontrolador.

Tabla 3.1 Pines de la Memoria MMC

- **!CS** permite al controlador seleccionar la memoria sobre la cual quiere operar, así cuando CS es igual a 0 la memoria se encuentra seleccionada y lista para operar. Este pin puede ser controlado por cualquier pin de salida del controlador.
- **DataIn** es la entrada de datos serie a la memoria y debe estar conectada a la salida MOSI de la interfaz SPI del controlador.
- **DataOut** es la salida de datos serie de la memoria y debe estar conectada al pin MISO de la interfaz SPI del controlador.
- **CLCK** es la señal de reloj generada por el controlador y es la que marca el ritmo de transferencia de la información serie entre ambos, así los datos se capturan o transmiten por la tarjeta al ritmo marcado por esta señal.

- **VDD** es el pin de alimentación y **VSS1** y **VSS2** son los pines de suministro de voltaje de tierra. Véase la tabla 3.1.

Tanto la tensión de alimentación como las señales deben encontrarse en el rango de los 2,6V y 3,7V.

3.1.2.1.5 Tarjeta de Memoria SD

Las memorias SD son muy similares a las MultiMediaCard a todos los niveles. No obstante, la primera diferencia que salta a la vista es el número de pines: las tarjetas SD tienen 9 pines, mientras que las MMC 7.

Esto es debido a que las memorias SD disponen de otros modos de acceso y protocolos que mejoran la seguridad en el intercambio de datos. No obstante a estas también se puede acceder mediante el mismo protocolo SPI que el utilizado en las MMC. Véase la figura 3.11.



Figura 3.11 Tarjeta de Memoria SD

3.1.2.1.5.1 Conexiones y Señales

Cuando se opera en modo SPI los pines de la memoria SD se denominan:

# Pin:	Nombre:	Tipo:	Descripción SPI:
9	DAT2	Reservado	No utilizado en modo SPI
1	!CD (!CS)	Entrada	Chip Select (activo a 0)
2	CMD (DataIn)	Entrada	Línea de datos y comandos hacia la memoria.
3	VSS1	Alimentación	Suministro de voltaje de tierra
4	VDD	Alimentación	Alimentación
5	CLCK	Entrada	Clock
6	VSS2	Alimentación	Suministro de voltaje de tierra
7	DAT0 (DataOut)	Salida	Línea de datos y status de la memoria al microcontrolador.
8	DAT1	Reservado	No utilizado en modo SPI

Tabla 3.2 Pines de la Memoria SD

- **!CD** (equivalente al !CS de la MMC) permite al controlador seleccionar la memoria sobre la cual quiere operar, así cuando CD vale 0 la tarjeta se encuentra seleccionada y lista para operar. Este pin puede ser controlado por cualquier pin de salida del controlador.
- **CMD** (equivalente al DataIn de la MMC) es la entrada de datos serie a la memoria y debe estar conectada a la salida MOSI de la interfaz SPI del controlador.

- **DAT0** (equivalente al DataOut de la MMC) es la salida de datos serie de la memoria y debe estar conectada al pin MISO de la interfaz SPI del controlador.
- **CLCK** es la señal de reloj generada por el controlador y es la que marca el ritmo de transferencia de la información serie entre ambos, así los datos se capturan o transmiten por la memoria al ritmo marcado por esta señal.
- **VDD** es el pin de alimentación y **VSS1** y **VSS2** son los pines de suministro de voltaje de tierra.
- **DAT1** y **DAT2** no se utilizan en el modo SPI

La alimentación y las señales deben encontrarse en el rango de los 2,6V y 3,7V.

3.1.2.1.6 MAX232CPE. Transceiver RS232

Durante el proceso de simulación fue necesario incorporar al sistema, una etapa de conversión de los niveles de salida de la tarjeta GPS, de 5V a 10V para así poder comunicarse con la PC y adquirir los datos emitidos por el GPS, de esta manera resultaría muy conveniente poder observar y procesar las tramas antes de realizar la implementación física.

Para este propósito se utilizó un transceiver (emisor y receptor) RS232 de la serie MAX232CPE. Este circuito integrado de 16 pines

en formato DIP, es capaz de supervisar 2 puertos seriales. Dentro de su circuitería interna y con la ayuda de 4 capacitores electrolíticos conectados en la configuración que se muestra en la figura 3.12 permitió establecer un vínculo entre la tarjeta GPS y la PC, teniendo así varias alternativas para visualizar las tramas y poder interpretarlas.

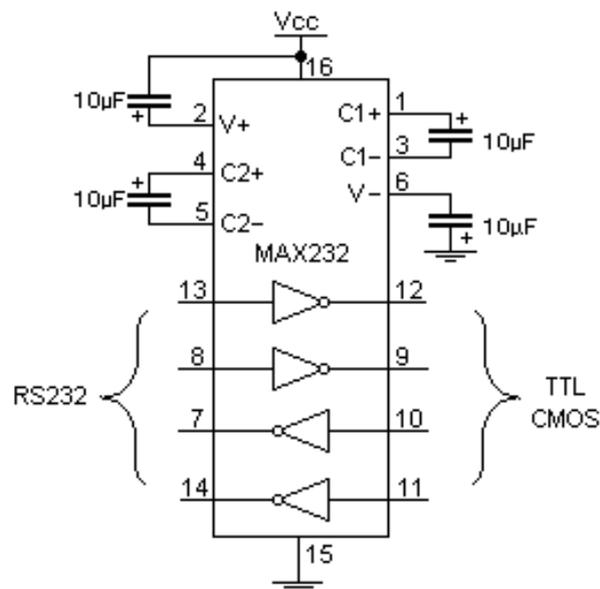


Figura 3.12 Configuración del MAX232

3.1.2.1.7 El dsPIC

Los dsPICs son el penúltimo lanzamiento de Microchip, comenzando a producirlos a gran escala a finales de 2004. Son los primeros PICs con bus de datos inherente de 16 bits; incorporan todas las posibilidades de los anteriores PICs y añaden varias

operaciones de DSP implementadas en hardware, como multiplicación con suma de acumulador (multiply-accumulate, o MAC), barrel shifting, bit reversion o multiplicación 16x16 bits.

En la figura 3.13 se muestra un ejemplo de un dsPIC.



Figura 3.13 El dsPIC

3.1.2.1.8 Sistema de Desarrollo dsPIC PRO4

El dsPICPRO4 soporta los últimos 64 y 80 pines dsPIC30F MCU de Microchip. Viene con una dsPIC30F6014A. El sistema incluye un programador de USB 2.0, mikroICD (In-Circuit del depurador), véase la figura 3.14.

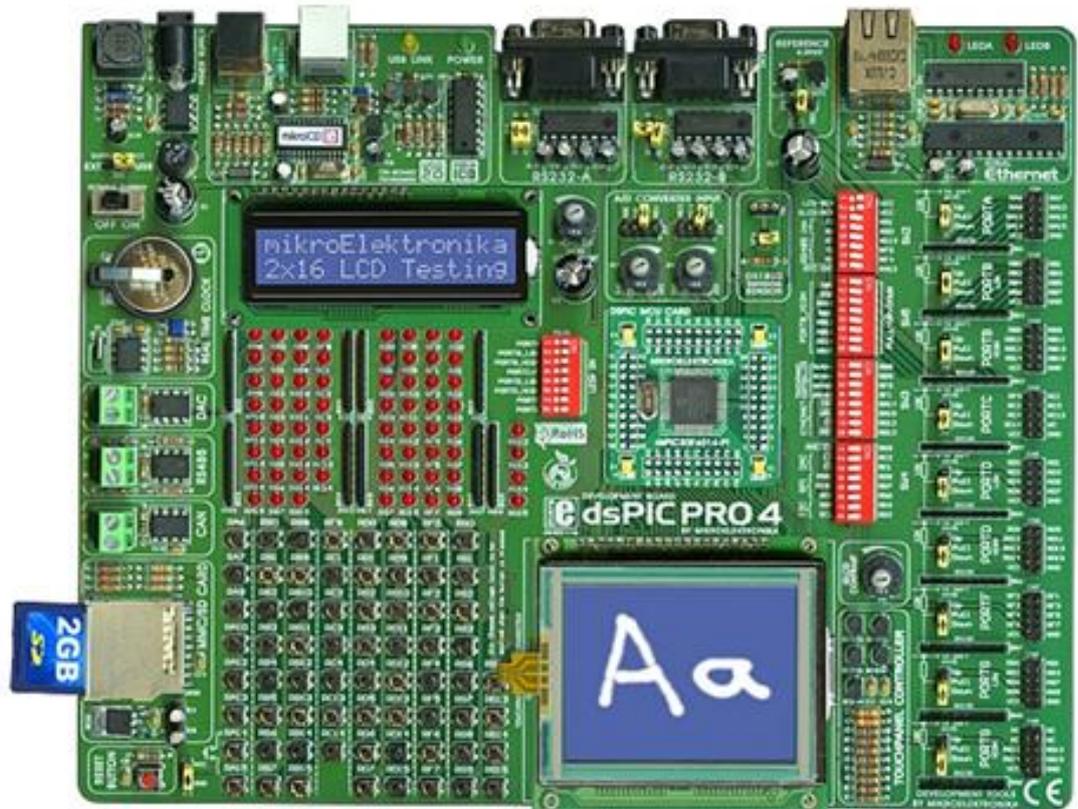


Figura 3.14 Sistema de Desarrollo dsPIC PRO4

EL dsPICPRO4 puede utilizar dos opciones de fuente de voltaje que son el USB, que es regulada por la PC y el de una fuente EXTERNA que puede ser de 9v hasta 32v, ya que el dsPICPRO4 utiliza un regulador interno MC34063A para producir 5v.

Se debe colocar el jumper J10 de acuerdo a la fuente de poder a la cual se desea trabajar, como se muestra en la figura 3.15.

En nuestro caso utilizamos una fuente externa de 9v, por lo que colocamos el jumper J10 al lado izquierdo.



Figura 3.15 Selección de Fuente de Poder

Para realizar la programación del dsPICPRO4 se utiliza el puerto USB 2.0, Véase la figura 3.16. Debemos conectar el cable USB a la PC, luego se debe cargar el programa del microcontrolador DSPIC30F6014A realizado en el software MikroC for dsPIC al software programador dsPicflash.



Figura 3.16 Programador USB 2.0

Para trabajar con la comunicación serial RS232 el dsPICPRO4 utiliza el MAX232 ya que los niveles de voltajes recibidos no son directamente compatibles, se deben colocar los jumpers J12 para

seleccionar la línea de recepción Rx que proviene del pin RF2 y el jumper J13 para seleccionar la línea de transmisión Tx proveniente del pin RF3.

Este proceso se realiza para tomar los datos procesados del microcontrolador 18F4520 con el puerto RS232-A. Véase la figura 3.17.

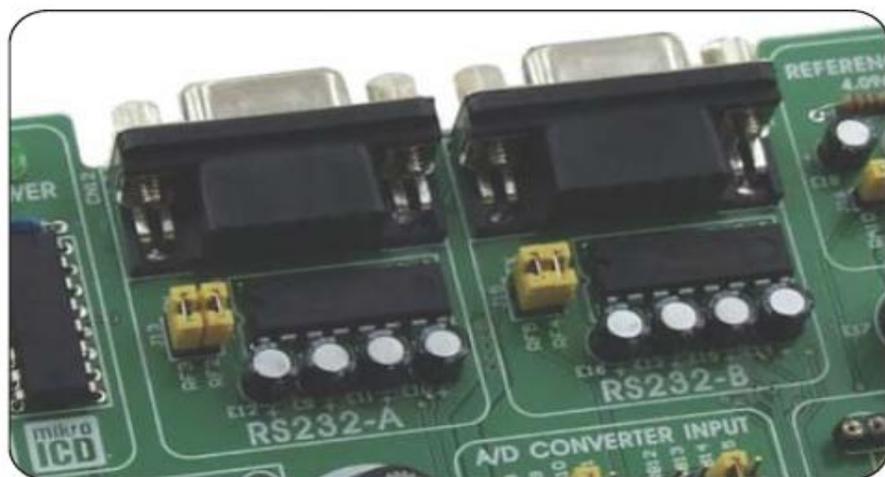


Figura 3.17 Comunicación RS232

Para trabajar con la comunicación SPI para guardar los datos en la memoria MMC/SD se colocaron los switches 1, 4, 5 y 6 del switch 4, para habilitar las líneas CS, MOSI, MISO y SCK. El dsPICPRO4 un regulador de voltaje MC33269DT.3.3 para convertir de 5v a 3.3v, debido a que la memoria MMC/SD trabaja a 3.3v. Se debe colocar la memoria en el módulo MMC/SD del dsPICPRO4. Véase la figura 3.18.

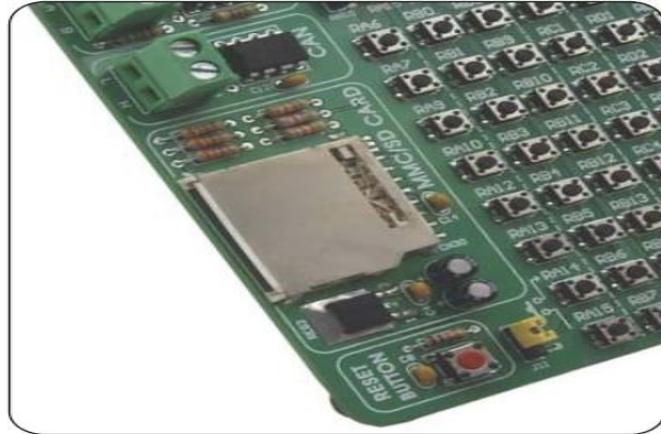


Figura 3.18 Tarjeta MMC/SD

CAPÍTULO 4

4 Simulación y Pruebas Experimentales

En este capítulo se presentan las diferentes simulaciones, pruebas y resultados que se realizaron en la fase de experimentación del proyecto.

4.1 Verificación de la Transmisión de Datos del Módulo GPS

Una vez obtenida la tarjeta SmartGPS era necesario investigar el funcionamiento de la misma, por lo que se realizaron lecturas de los datos que envía la tarjeta con programas diferentes.

En esta prueba intervinieron los siguientes componentes:

- Tarjeta SmartGPS
- Convertidor de TTL a RS232
- Computador (Puerto Serial)

Usando los pines TX, VCC y GND de la tarjeta SmartGPS se lo conecta al Convertidor de TTL a RS232 y el pin TX del convertidor de TTL a RS232 al RX del puerto serial de la computadora.

4.1.1 HyperTerminal

Se accede al programa HyperTerminal, el cual se encuentra en las computadoras que tienen instalado el Windows XP. Se procede a configurar el HyperTerminal a una tasa de baudios de 4800 y se selecciona el puerto COM correspondiente al puerto serial, luego se alimenta a la tarjeta SmartGPS con 9V hasta obtener lecturas del GPS, tal y como se muestra en la figura 4.1. Esta prueba con el HyperTerminal sólo se puede realizar con Windows XP.

```

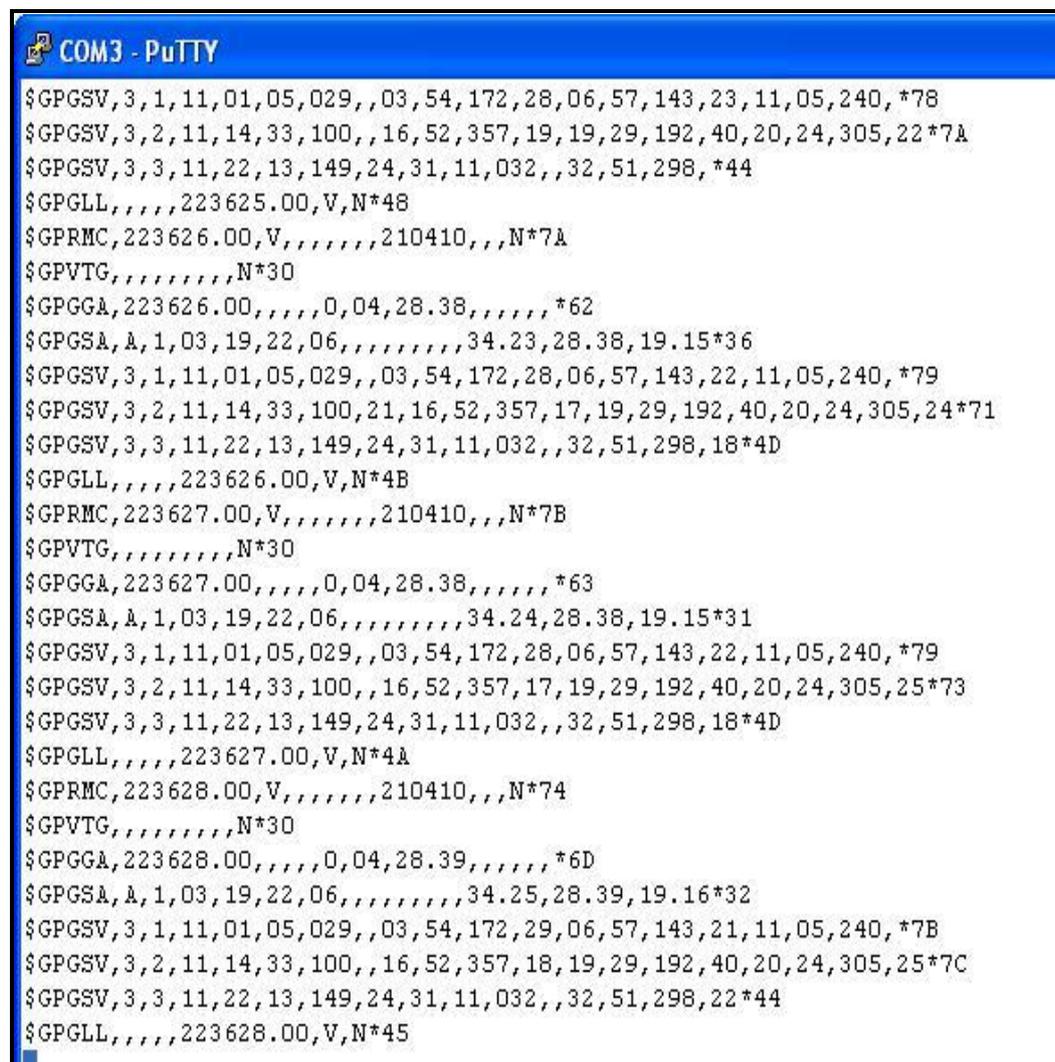
gps - HyperTerminal
Archivo Edición Ver Llamar Transferir Ayuda
$GPVTG, , T, , M, 0.161, N, 0.298, K, A*26
$GPGGA, 231035.00, 0208.69068, S, 07958.07253, W, 1.05, 2.57, 78.5, M, 10.8, M, , *68
$GPGSA, A, 3, 19, 32, 03, 20, 06, , , , , , 5.09, 2.57, 4.40*00
$GPGSV, 3, 1, 11, 03, 66, 156, 35, 06, 64, 115, 21, 11, 08, 229, , 14, 29, 116, *77
$GPGSV, 3, 2, 11, 16, 39, 004, , 19, 40, 184, 46, 20, 31, 293, 25, 22, 03, 146, *7D
$GPGSV, 3, 3, 11, 23, 08, 333, , 31, 11, 044, , 32, 57, 274, 33*43
$GPRML, 0208.69068, S, 07958.07253, W, 231035.00, A, A*6B
$GPRMC, 231036.00, A, 0208.69077, S, 07958.07270, W, 0.164, , 200410, , , A*7A
$GPVTG, , T, , M, 0.164, N, 0.305, K, A*26
$GPGGA, 231036.00, 0208.69077, S, 07958.07270, W, 1.05, 2.57, 78.6, M, 10.8, M, , *67
$GPGSA, A, 3, 19, 32, 03, 20, 06, , , , , , 5.09, 2.57, 4.40*00
$GPGSV, 3, 1, 11, 03, 66, 156, 35, 06, 64, 115, 22, 11, 08, 229, , 14, 29, 116, *74
$GPGSV, 3, 2, 11, 16, 39, 004, , 19, 40, 184, 46, 20, 31, 293, 25, 22, 03, 146, *7D
$GPGSV, 3, 3, 11, 23, 08, 333, , 31, 11, 044, , 32, 57, 274, 32*42
$GPRML, 0208.69077, S, 07958.07270, W, 231036.00, A, A*67
$GPRMC, 231037.00, A, 0208.69085, S, 07958.07289, W, 0.033, , 200410, , , A*73
$GPVTG, , T, , M, 0.033, N, 0.061, K, A*24
$GPGGA, 231037.00, 0208.69085, S, 07958.07289, W, 1.05, 2.57, 78.7, M, 10.8, M, , *6C
$GPGSA, A, 3, 19, 32, 03, 20, 06, , , , , , 5.09, 2.57, 4.40*00
$GPGSV, 3, 1, 11, 03, 66, 156, 35, 06, 64, 115, 23, 11, 08, 229, , 14, 29, 116, *75
$GPGSV, 3, 2, 11, 16, 39, 004, , 19, 40, 184, 46, 20, 31, 293, 24, 22, 03, 146, *7C
$GPGSV, 3, 3, 11, 23, 08, 333, , 31, 11, 044, , 32, 57, 274, 31*41
$GPRML, 0208.69085, S, 07958.07289, W, 231037.00, A, A*6D
0:00:15 conectado Autodetect. 9600 8-N-1 DESPLAZAR MAY NUM Capturar Imprimir

```

Figura 4.1 Prueba con el HyperTerminal

4.1.2 Programa PUTTY

Se ejecuta la aplicación PUTTY, se configura tipo de conexión a serial y se escoge el puerto COM donde está conectado el cable serial, luego se alimenta la tarjeta SmartGPS a 9V hasta obtener las lecturas del GPS tal y como se muestra en la figura 4.2.



```

COM3 - PuTTY
$GPGSV,3,1,11,01,05,029,,03,54,172,28,06,57,143,23,11,05,240,*78
$GPGSV,3,2,11,14,33,100,,16,52,357,19,19,29,192,40,20,24,305,22*7A
$GPGSV,3,3,11,22,13,149,24,31,11,032,,32,51,298,*44
$GPGLL,,,,,223625.00,V,N*48
$GPRMC,223626.00,V,,,,,,210410,,,N*7A
$GPVTG,,,,,,N*30
$GPGGA,223626.00,,,,,0,04,28.38,,,,,*62
$GPGSA,A,1,03,19,22,06,,,,,,34.23,28.38,19.15*36
$GPGSV,3,1,11,01,05,029,,03,54,172,28,06,57,143,22,11,05,240,*79
$GPGSV,3,2,11,14,33,100,21,16,52,357,17,19,29,192,40,20,24,305,24*71
$GPGSV,3,3,11,22,13,149,24,31,11,032,,32,51,298,18*4D
$GPGLL,,,,,223626.00,V,N*4B
$GPRMC,223627.00,V,,,,,,210410,,,N*7B
$GPVTG,,,,,,N*30
$GPGGA,223627.00,,,,,0,04,28.38,,,,,*63
$GPGSA,A,1,03,19,22,06,,,,,,34.24,28.38,19.15*31
$GPGSV,3,1,11,01,05,029,,03,54,172,28,06,57,143,22,11,05,240,*79
$GPGSV,3,2,11,14,33,100,,16,52,357,17,19,29,192,40,20,24,305,25*73
$GPGSV,3,3,11,22,13,149,24,31,11,032,,32,51,298,18*4D
$GPGLL,,,,,223627.00,V,N*4A
$GPRMC,223628.00,V,,,,,,210410,,,N*74
$GPVTG,,,,,,N*30
$GPGGA,223628.00,,,,,0,04,28.39,,,,,*6D
$GPGSA,A,1,03,19,22,06,,,,,,34.25,28.39,19.16*32
$GPGSV,3,1,11,01,05,029,,03,54,172,29,06,57,143,21,11,05,240,*7B
$GPGSV,3,2,11,14,33,100,,16,52,357,18,19,29,192,40,20,24,305,25*7C
$GPGSV,3,3,11,22,13,149,24,31,11,032,,32,51,298,22*44
$GPGLL,,,,,223628.00,V,N*45

```

Figura 4.2 Prueba con el Programa PUTTY

4.1.3 Terminal Virtual del Proteus

Se implementa una pequeña simulación con una herramienta denominada COMPIM y un terminal virtual, se configura el COMPIM para configurar el puerto físico con el puerto COM donde está conectado el cable serial, luego se energiza la tarjeta SmartGPS, para de esta manera obtener las lecturas del GPS tal como se muestra en la figura 4.3.

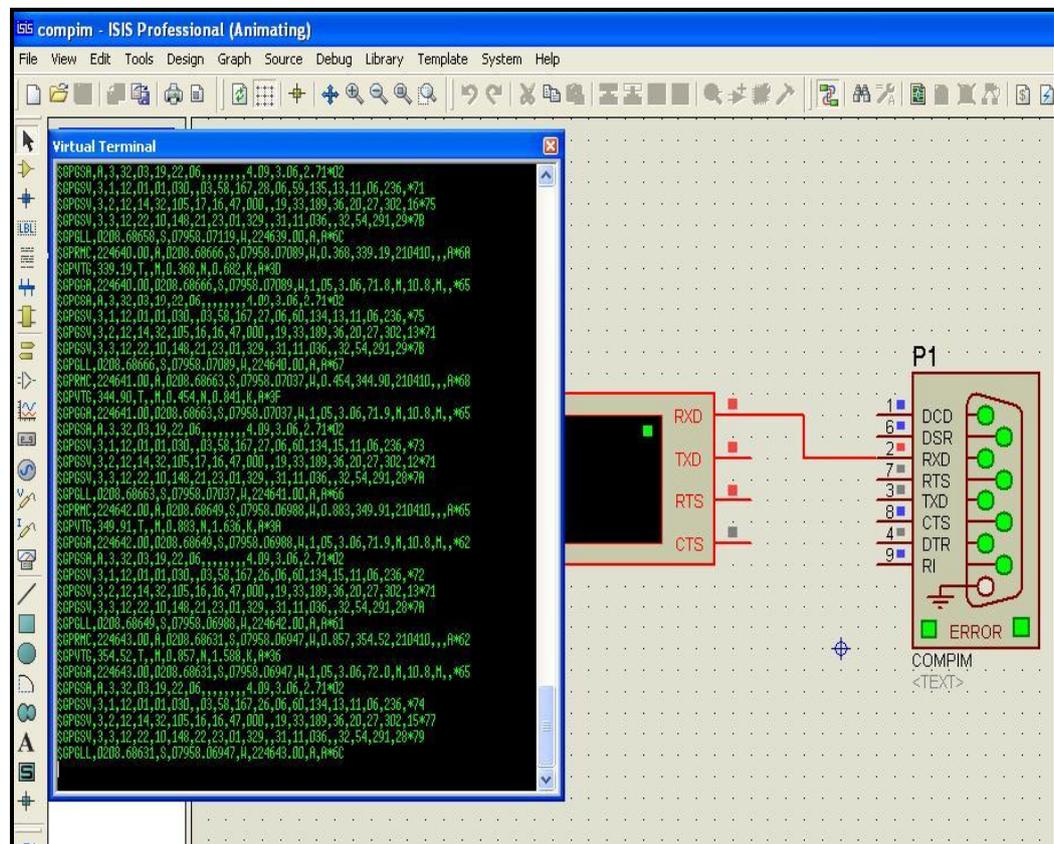


Figura 4.3 Prueba realizada con Proteus usando el Virtual Terminal

4.2 Simulaciones Realizadas

4.2.1 Ejemplo MMC_FAT_16 en Proteus

Para poner en funcionamiento el ejemplo que se encuentra en las librerías que proporciona el compilador MikroBasic PRO for PIC se realizó la simulación en Proteus del circuito que se muestra en la figura 4.4; luego se creó el proyecto CodigoMMC usando MikroBasic y tomando las líneas de código que se incluyen en este archivo, se compila el programa y este se lo carga en el PIC 18F4520 de la simulación en Proteus.

El sistema proporciona mediante un terminal virtual colocado en la simulación un mensaje de inicio (START), que indica que la Card Multimedia esta lista para grabar datos. Una vez concluido el almacenamiento de los datos en la Card Multimedia se presenta un mensaje de finalización (END).

Para realizar correctamente la ejecución del programa se debe crear un archivo virtual con extensión mcif y colocarlo en la tarjeta MMC de la simulación. Por lo tanto se modifica el código del ejemplo MMC_FAT_16, para que el programa únicamente se encargue de escribir los datos sobre el archivo que se simuló y que hará las veces del archivo que se creará en la memoria MMC/SD una vez que el proyecto se ejecute en la realidad.

Para revisar los datos guardados se procede a buscar la ruta en donde se encuentra el archivo con extensión mcif, el cual se lo denominó PRUEBAS. El resultado de la simulación se muestra en la figura 4.5.

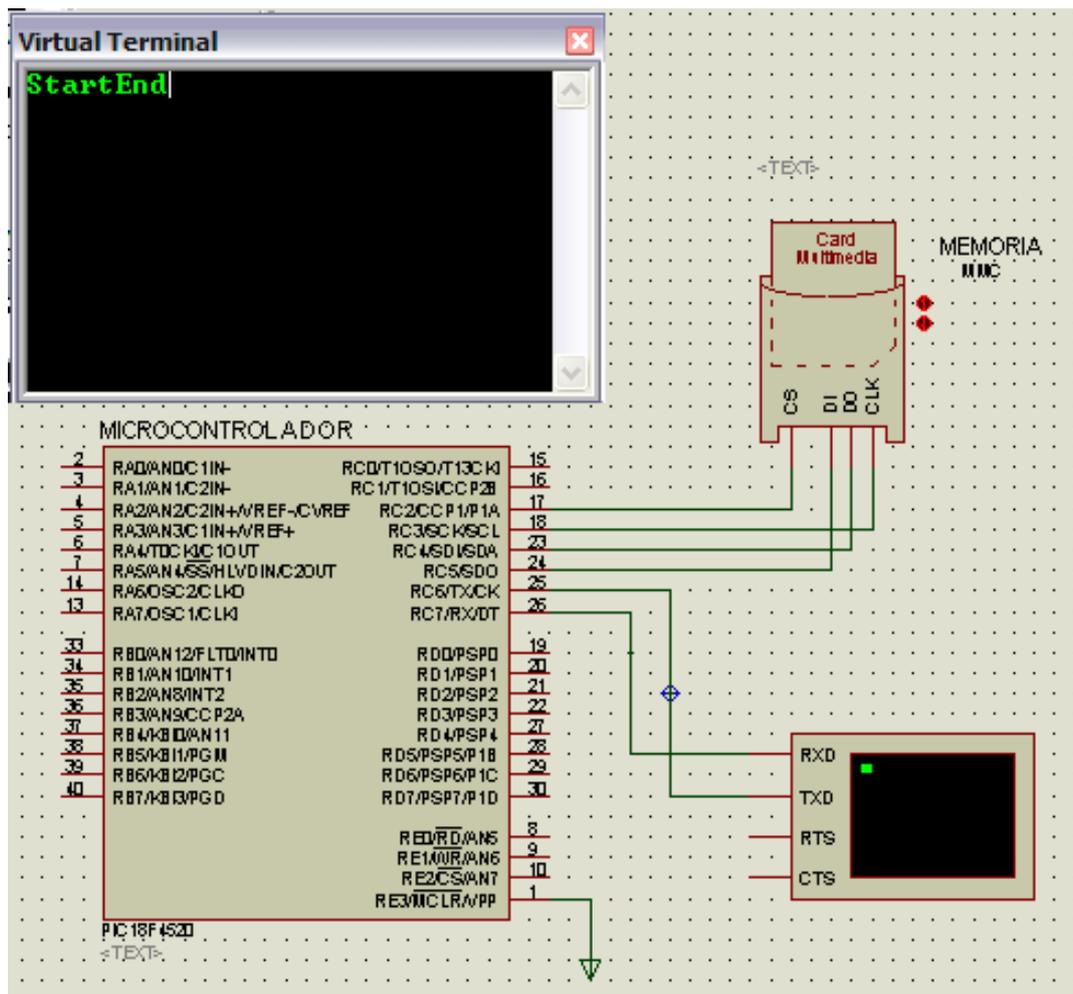


Figura 4.4 Simulación del Ejemplo MMC_FAT_16

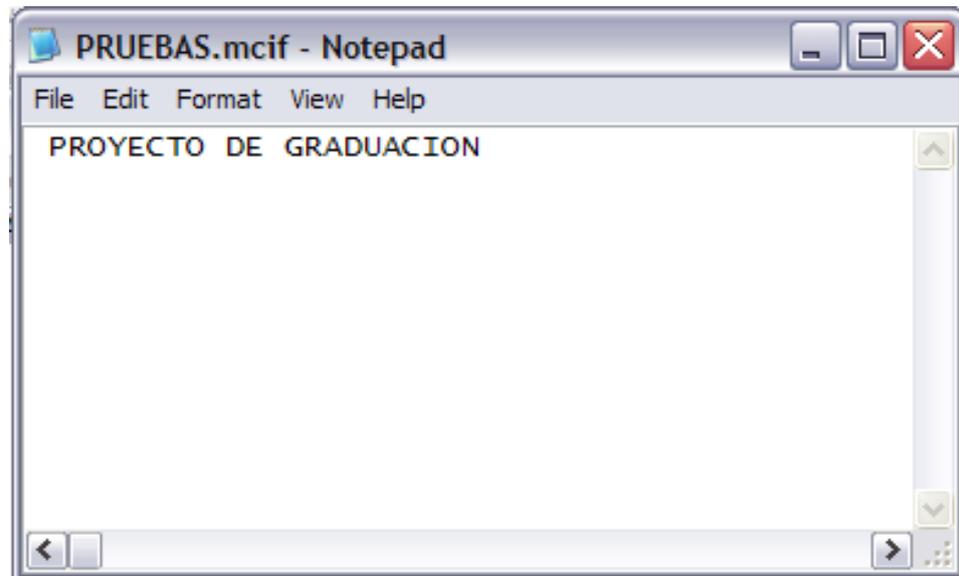


Figura 4.5 Archivo Grabado en la Simulación

4.2.2 Proyecto en Proteus

Para ejecutar el proyecto en Proteus se requiere que además de ensamblar el circuito tal y como se muestra en la figura 4.4 se utilice el COMPIM; que es una herramienta propia del software Proteus, la cual servirá para configurar un puerto físico donde se conecte la tarjeta SmartGPS a través del convertidor de TTL a RS232 y el puerto serial de la computadora por ejemplo COM1, luego se procede a alimentar la tarjeta a 9V. La simulación debe presentarse como se muestra en la figura 4.6.

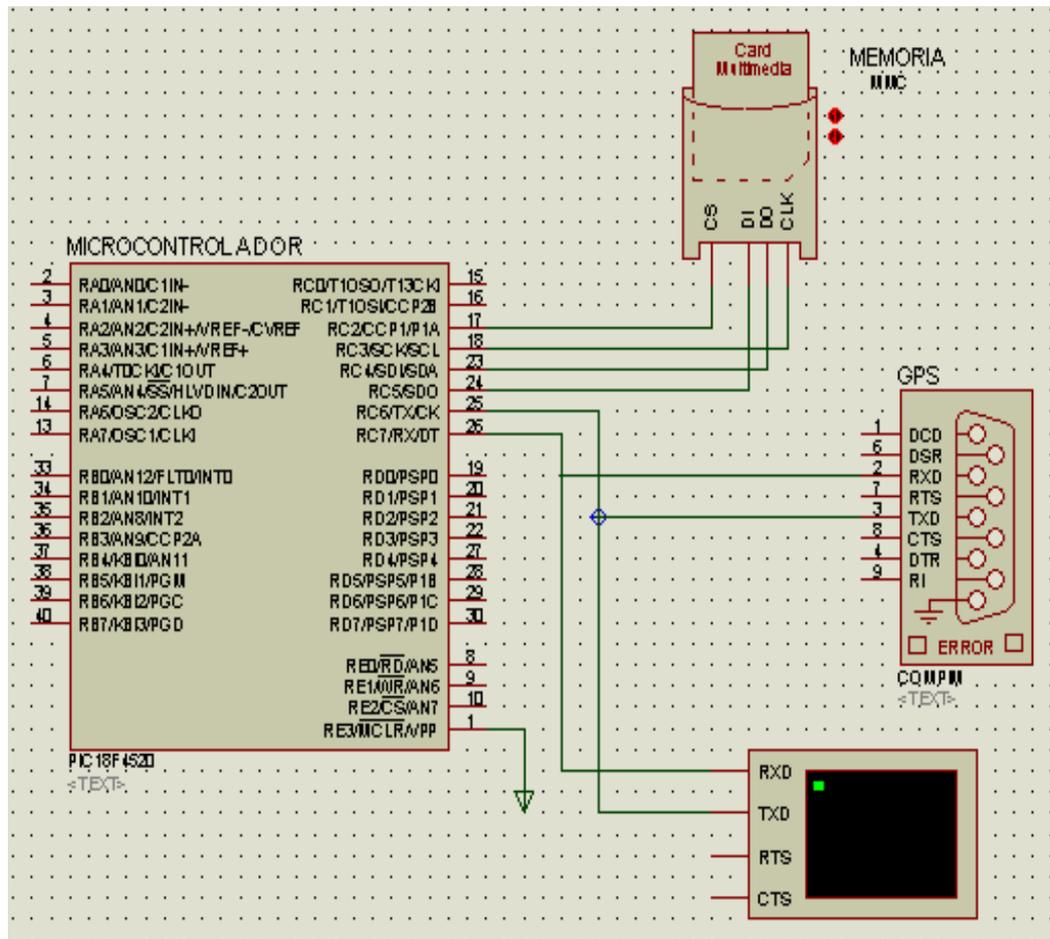


Figura 4.6 Simulación del Proyecto en Proteus

4.2.2.1 Resultados de la Simulación del Proyecto

Una vez finalizada la ejecución de la simulación se revisa el archivo virtual que se guardó en la Card Multimedia. El resultado procede como se muestra en la figura 4.7.

El archivo contiene un conjunto de tramas pertenecientes al protocolo NMEA, específicamente corresponden al grupo de cadenas GPGGA; el cual abarca 15 campos provistos por un conjunto de

satélites con información sobre posicionamiento, orientación, hora, altura, etc.

El contenido correspondiente al Protocolo NMEA se encuentra desglosado en detalle en el capítulo 2.

Archivo	Edición	Formato	Ver	Ayuda
\$GPGGA,234043.00,0211.47755,S,07953.64499,W,1,04,9.46,-3.1,M,10.8,M,,*70				
\$GPGGA,234044.00,0211.47616,S,07953.64647,W,1,05,9.46,-2.8,M,10.8,M,,*79				
\$GPGGA,234045.00,0211.47565,S,07953.64671,W,1,06,9.45,-2.9,M,10.8,M,,*7B				
\$GPGGA,234046.00,0211.47442,S,07953.64838,W,1,06,3.24,-2.4,M,10.8,M,,*7F				
\$GPGGA,234047.00,0211.47608,S,07953.64468,W,1,06,2.05,-4.1,M,10.8,M,,*7A				
\$GPGGA,234048.00,0211.47737,S,07953.64196,W,1,06,2.05,-5.4,M,10.8,M,,*78				
\$GPGGA,234049.00,0211.47821,S,07953.64024,W,1,06,2.05,-6.3,M,10.8,M,,*7D				
\$GPGGA,234050.00,0211.47910,S,07953.63833,W,1,06,2.05,-7.0,M,10.8,M,,*7D				
\$GPGGA,234051.00,0211.47945,S,07953.63759,W,1,06,2.05,-7.2,M,10.8,M,,*7D				
\$GPGGA,234052.00,0211.47976,S,07953.63688,W,1,06,2.05,-7.3,M,10.8,M,,*72				
\$GPGGA,234053.00,0211.47994,S,07953.63645,W,1,06,2.05,-7.1,M,10.8,M,,*7C				
\$GPGGA,234054.00,0211.47995,S,07953.63656,W,1,06,2.05,-7.2,M,10.8,M,,*7B				
\$GPGGA,234055.00,0211.48013,S,07953.63623,W,1,06,2.05,-7.2,M,10.8,M,,*70				
\$GPGGA,234056.00,0211.48042,S,07953.63571,W,1,06,2.05,-7.3,M,10.8,M,,*72				
\$GPGGA,234057.00,0211.48074,S,07953.63536,W,1,06,2.05,-7.5,M,10.8,M,,*73				
\$GPGGA,234058.00,0211.48101,S,07953.63497,W,1,06,2.05,-7.6,M,10.8,M,,*76				
\$GPGGA,234059.00,0211.48128,S,07953.63444,W,1,06,2.05,-7.9,M,10.8,M,,*7D				
\$GPGGA,234100.00,0211.48152,S,07953.63413,W,1,06,2.05,-8.2,M,10.8,M,,*7B				
\$GPGGA,234101.00,0211.48186,S,07953.63362,W,1,06,2.05,-8.5,M,10.8,M,,*75				
\$GPGGA,234102.00,0211.48199,S,07953.63355,W,1,06,2.05,-8.7,M,10.8,M,,*7E				
\$GPGGA,234103.00,0211.48201,S,07953.63326,W,1,06,2.05,-8.9,M,10.8,M,,*77				
\$GPGGA,234104.00,0211.48205,S,07953.63308,W,1,06,2.04,-9.2,M,10.8,M,,*73				
\$GPGGA,234105.00,0211.48212,S,07953.63303,W,1,06,2.04,-9.4,M,10.8,M,,*79				
\$GPGGA,234106.00,0211.48208,S,07953.63325,W,1,06,2.04,-9.5,M,10.8,M,,*74				
\$GPGGA,234107.00,0211.48212,S,07953.63343,W,1,06,2.04,-9.4,M,10.8,M,,*7F				
\$GPGGA,234108.00,0211.48219,S,07953.63379,W,1,06,2.04,-9.4,M,10.8,M,,*72				
\$GPGGA,234109.00,0211.48238,S,07953.63378,W,1,06,2.04,-9.4,M,10.8,M,,*71				
\$GPGGA,234110.00,0211.48249,S,07953.63388,W,1,06,2.04,-9.4,M,10.8,M,,*70				
\$GPGGA,234111.00,0211.48250,S,07953.63418,W,1,06,2.04,-9.4,M,10.8,M,,*77				
\$GPGGA,234112.00,0211.48239,S,07953.63464,W,1,06,2.04,-9.3,M,10.8,M,,*77				
\$GPGGA,234113.00,0211.48230,S,07953.63507,W,1,06,2.04,-9.2,M,10.8,M,,*7A				
\$GPGGA,234114.00,0211.48232,S,07953.63537,W,1,06,2.04,-9.1,M,10.8,M,,*7F				
\$GPGGA,234115.00,0211.48264,S,07953.63501,W,1,06,2.04,-9.0,M,10.8,M,,*79				
\$GPGGA,234116.00,0211.48280,S,07953.63494,W,1,06,2.04,-9.0,M,10.8,M,,*7D				
\$GPGGA,234117.00,0211.48284,S,07953.63529,W,1,06,2.04,-8.9,M,10.8,M,,*77				
\$GPGGA,234118.00,0211.48294,S,07953.63529,W,1,06,2.04,-8.8,M,10.8,M,,*78				
\$GPGGA,234119.00,0211.48314,S,07953.63529,W,1,06,2.04,-8.8,M,10.8,M,,*70				

Figura 4.7 Resultados de la Simulación

4.3 Implementación del Proyecto

En la figura 4.8 se pone en manifiesto la estructura que conforma el prototipo que se diseñó con la finalidad de realizar la tarea de almacenamiento masivo de sentencias NMEA.

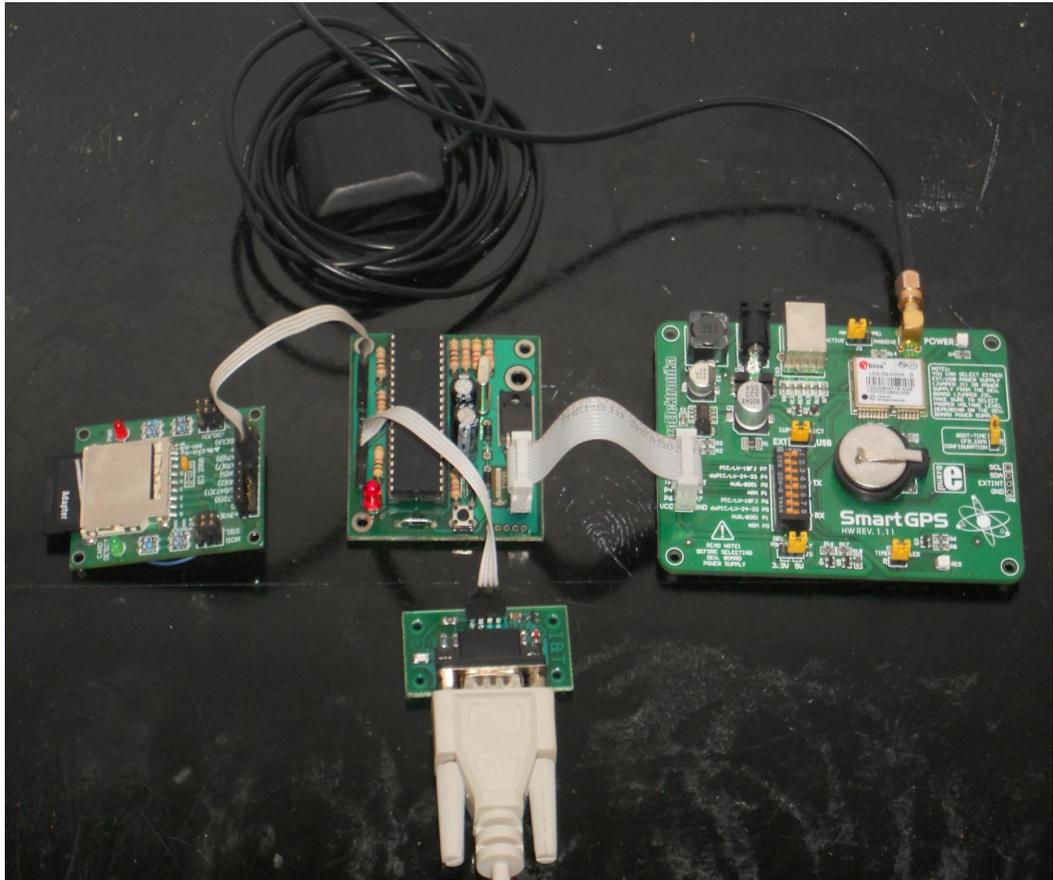


Figura 4.8 Implementación Física del Proyecto

4.3.1 Resultados de la Implementación del Proyecto

El archivo que se crea al poner en funcionamiento el sistema es llamado DATOSGPS; este representa el resultado final de la implementación del proyecto.

En una primera instancia se puede apreciar en la figura 4.9 tramas de inicialización, pues el módulo GPS aún no se ha contactado con los satélites que requiere para su correcto funcionamiento.

Las pruebas con la tarjeta SmartGPS se la realizó tomando puntos fijos, debido a que en el momento de desarrollar las respectivas pruebas, el prototipo no estaba en movimiento; por lo tanto las tramas de las lecturas presentadas en la figura 4.9 contienen coordenadas similares en longitud y latitud. Sin embargo el sistema está diseñado para poder ser utilizado en movimiento, de tal manera que su aplicación se enfoque en situaciones reales y no meramente académicas.

Como dato adicional se determinó que el módulo GPS presenta un rango de error aceptado de aproximadamente 5 metros a la redonda.

CONCLUSIONES Y RECOMENDACIONES

1. Al realizar este proyecto se pudo establecer la gran utilidad que presenta el empleo de los sistemas de posicionamiento global vinculados con microcontroladores para poder llevar a cabo diversas aplicaciones.
2. Una vez finalizado el proyecto podemos concluir que la tarjeta de desarrollo utilizada fue muy adecuada para nuestro propósito tanto en prestaciones como en facilidades de uso, debido a la capacidad de comunicación con otros dispositivos y a las soluciones integradas que posee, como lo es su batería de respaldo, módulo de regulación de voltaje, la fácil identificación de sus puertos de conexión, etc.
3. Se puede concluir que para un correcto funcionamiento de la tarjeta ET-MINI SD/MMC es necesario inicializar la memoria MMC/SD, para esto se requirió colocar ciertos retardos para que la memoria logre ser inicializada adecuadamente antes de mandar a grabar los datos.
4. Se pudo concluir que el proyecto resulta de mucha utilidad puesto que su versatilidad radica en supervisar grandes cantidades de datos; que

no simplemente se enfrasca en almacenamiento de tramas, sino que puede ser aplicable en monitoreo de temperatura, de sensores, se señales, etc.

5. Mediante la realización del proyecto se aprendió que para poder grabar datos en la memoria MMC/SD, es imprescindible lograr inicializarla correctamente; para esto se requirió establecer un rango de reloj entre 300 y 400 KHz en el instante en que se configura la librería FAT.

Las recomendaciones son:

1. Es necesario que el usuario se familiarice primero tanto con la documentación de los componentes que forman parte de la tarjeta SmartGPS como de la tarjeta ET-MINI SD/MMC; ya que esto le ayudará en el uso de las mismas y así proteger la integridad de ambas tarjetas. Una vez que se tenga el conocimiento de los componentes que conforman las tarjetas por separado, es importante revisar la interacción entre todos y cada uno de los componentes.
2. Es importante verificar el funcionamiento de la tarjeta, esto se puede realizar comunicando el puerto serial que posee la tarjeta SmartGPS, un circuito con el micro MAX232 y una computadora con puerto serial, la computadora debe tener HyperTerminal o una aplicación como el PUTTY que permita visualizar la información del GPS.
3. El modo de alimentación de la tarjeta SmartGPS debe ser por una fuente externa, ya que esta tarjeta tiene un funcionamiento para cada tipo de alimentación, cuando la alimentación es por una fuente externa el módulo GPS comienza a transmitir los datos de la ubicación; cuando la alimentación es por USB el módulo GPS envía datos de

inicialización y entra en un estado de espera de datos de comunicación.

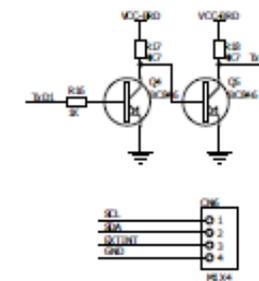
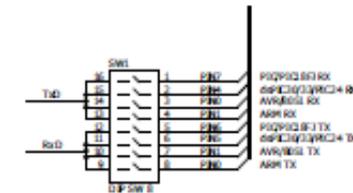
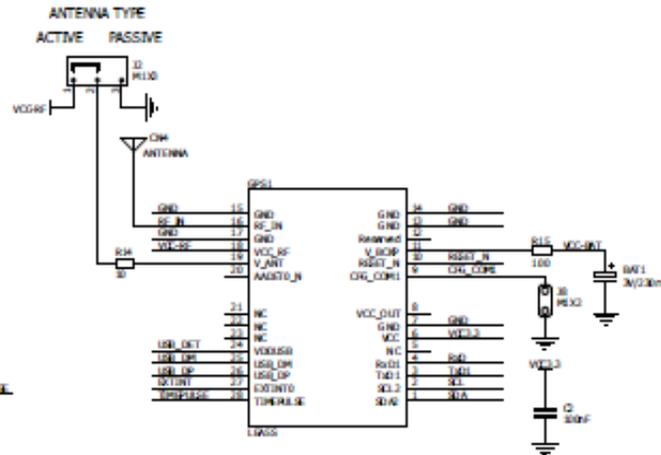
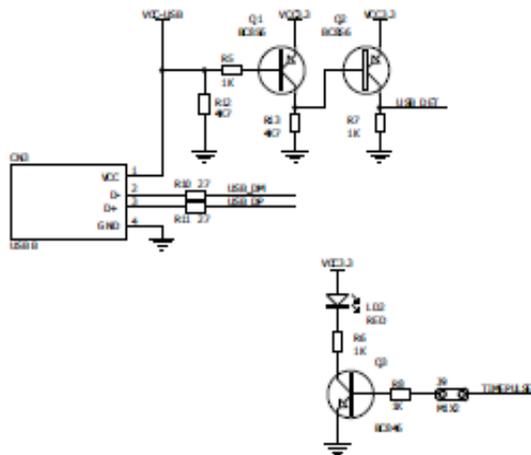
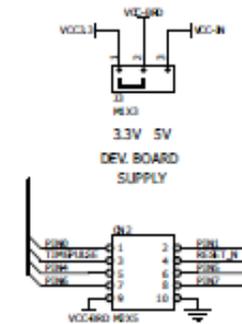
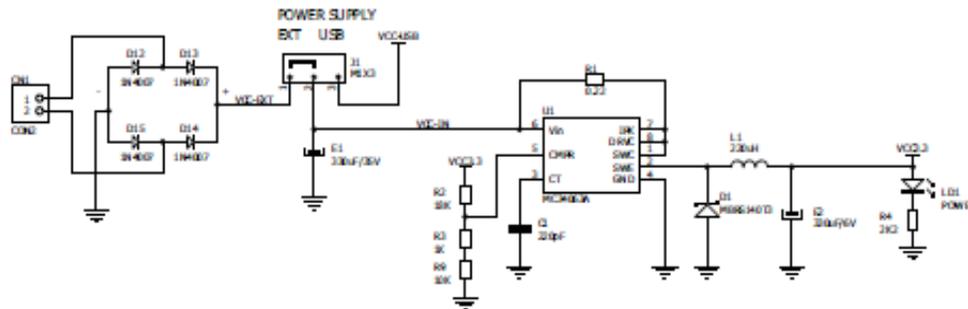
4. En la tarjeta ET-MINI SD/MMC las señales deben encontrarse en el rango de los 2,6V a 3,7V para que trabaje adecuadamente; por lo que es recomendable bajar el valor de los 5V que envía el microcontrolador a la tarjeta a través de resistencias.
5. El programa MikroBasic Pro for PIC resulta una herramienta muy práctica para el manejo de módulos GPS, ya que la implementación de una estructura para la lectura y presentación de datos es muy sencilla, además de que provee herramientas muy valiosas para realizar las correspondientes simulaciones.
6. Es recomendable investigar el funcionamiento del programa GoogleMaps o GoogleEarth, ya que esto ayuda a entender el funcionamiento del GPS y el manejo de las coordenadas para la representación gráfica de los datos que envía dicho módulo.

ANEXOS

ANEXO 1

Hoja de datos técnicos módulo GPS LEA5S

SWITCHING POWER SUPPLY



ANEXO 2

Hoja de datos técnicos PIC18F4520



MICROCHIP

PIC18F2420/2520/4420/4520

28/40/44-Pin Enhanced Flash Microcontrollers with 10-Bit A/D and nanoWatt Technology

Power Managed Modes:

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Idle mode currents down to 5.8 μ A typical
- Sleep mode current down to 0.1 μ A typical
- Timer1 Oscillator: 1.8 μ A, 32 kHz, 2V
- Watchdog Timer: 2.1 μ A
- Two-Speed Oscillator Start-up

Peripheral Highlights:

- High-current sink/source 25 mA/25 mA
- Three programmable external interrupts
- Four input change interrupts
- Up to 2 Capture/Compare/PWM (CCP) modules, one with Auto-Shutdown (28-pin devices)
- Enhanced Capture/Compare/PWM (ECCP) module (40/44-pin devices only):
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-Shutdown and Auto-Restart
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI™ (all 4 modes) and I²C™ Master and Slave Modes
- Enhanced Addressable USART module:
 - Supports RS-485, RS-232 and LIN 1.2
 - RS-232 operation using internal oscillator block (no external crystal required)
 - Auto-Wake-up on Start bit
 - Auto-Baud Detect
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual analog comparators with input multiplexing)

Flexible Oscillator Structure:

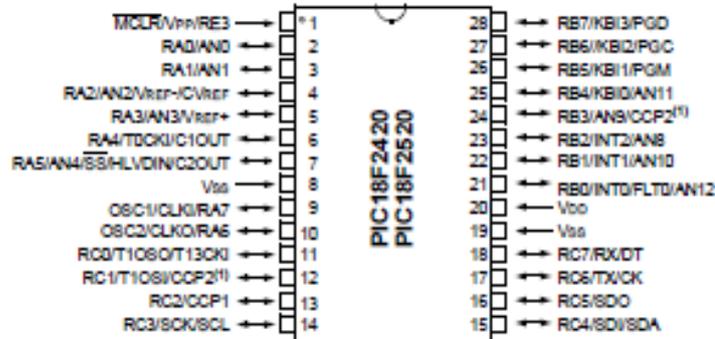
- Four Crystal modes, up to 40 MHz
- 4X Phase Lock Loop (available for crystal and internal oscillators)
- Two External RC modes, up to 4 MHz
- Two External Clock modes, up to 40 MHz
- Internal oscillator block:
 - 8 user selectable frequencies, from 31 kHz to 8 MHz
 - Provides a complete range of clock speeds from 31 kHz to 32 MHz when used with PLL
 - User tunable to compensate for frequency drift
- Secondary oscillator using Timer1 @ 32 kHz
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if peripheral clock stops

Special Microcontroller Features:

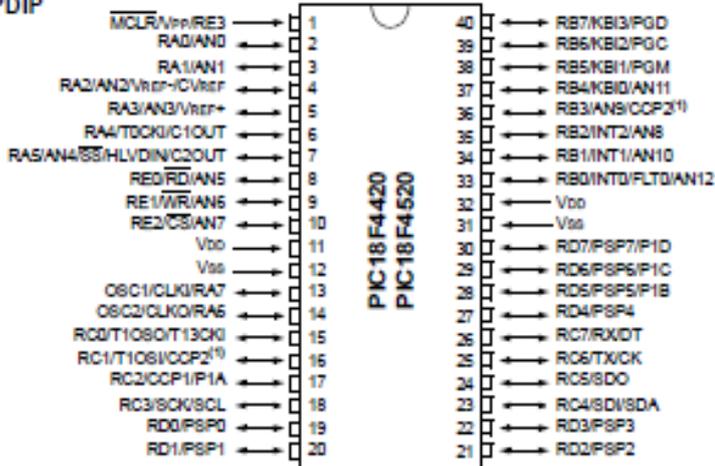
- C compiler optimized architecture:
 - Optional extended instruction set designed to optimize re-entrant code
- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- Flash/Data EEPROM Retention: 100 years typical
- Self-programmable under software control
- Priority levels for interrupts
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
 - Programmable period from 4 ms to 131s
- Single-supply 5V In-Circuit Serial Programming™ (ICSP™) via two pins
- In-Circuit Debug (ICD) via two pins
- Wide operating voltage range: 2.0V to 5.5V
- Programmable 16-level High/Low-Voltage Detection (HLVD) module:
 - Supports interrupt on High/Low-Voltage Detection
- Programmable Brown-out Reset (BOR)
 - With software enable option

Pin Diagrams

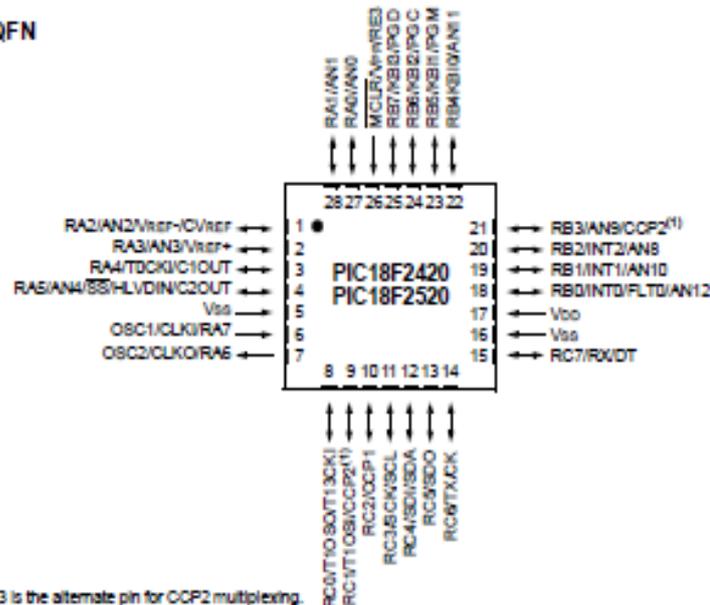
28-pin PDIP, SOIC



40-pin PDIP



28-pin QFN



Note 1: RB3 is the alternate pin for CCP2 multiplexing.

PIC18F2420/2520/4420/4520

Device	Program Memory		Data Memory		I/O	10-bit A/D (ch)	CCP/ ECCP (PWM)	MSSP		EUSART	Comp.	Timers 8/16-bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)				SPI	Master I ² C			
PIC18F2420	16K	8192	768	256	25	10	2/0	Y	Y	1	2	1/3
PIC18F2520	32K	16384	1536	256	25	10	2/0	Y	Y	1	2	1/3
PIC18F4420	16K	8192	768	256	36	13	1/1	Y	Y	1	2	1/3
PIC18F4520	32K	16384	1536	256	36	13	1/1	Y	Y	1	2	1/3

ANEXO 3

Hoja de datos técnicos ET-MINI SD/MMC

ET-MINI SD/MMC

The connection SD/MMC MEMORY CARD

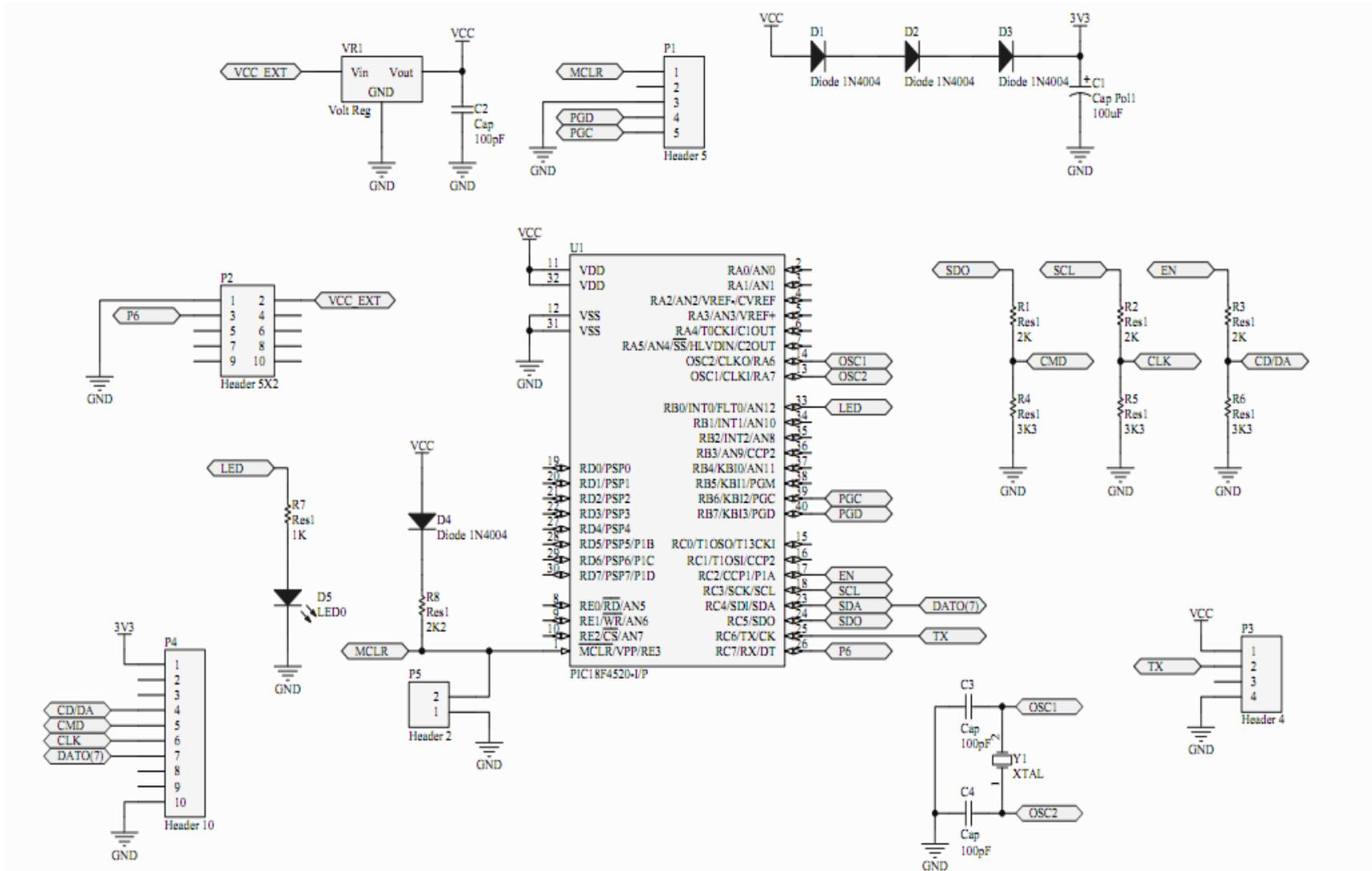
ET-MINI SD/MMC is a device for connection components with SD/MMC memory Card; for example, the connection between Microcontroller and Memory Card (SD/MMC). This set contains Socket for inserting Memory Card such as SD and MMC and it provides external signal Pin at Connector Pin, so it is quite convenient to apply. Moreover, there are many circuits such as Circuit Card Detect, Circuit Pull-Up and etc.

Specifications of ET-MINI SD/MMC

- Support Memory Card as SD type and MMC type
- Can select Enable/ Disable for Circuit Pull-Up
- Can display status of inserting Card or CARD DETECT by using LED and send signal OUTPUT through Pin signal CD.
There is specification as follows;
 CD = 1; It means that there is no Card.
 CD = 0; It means that there is Card.
- Can display status of Switch Write Protection on SD/MMC Card and send signal OUTPUT through signal Pin WP as follows;
 WP = 1; It is OFF position of Switch Write Protection.
 WR = 0; It is ON position of Switch Write Protection.

ANEXO 4

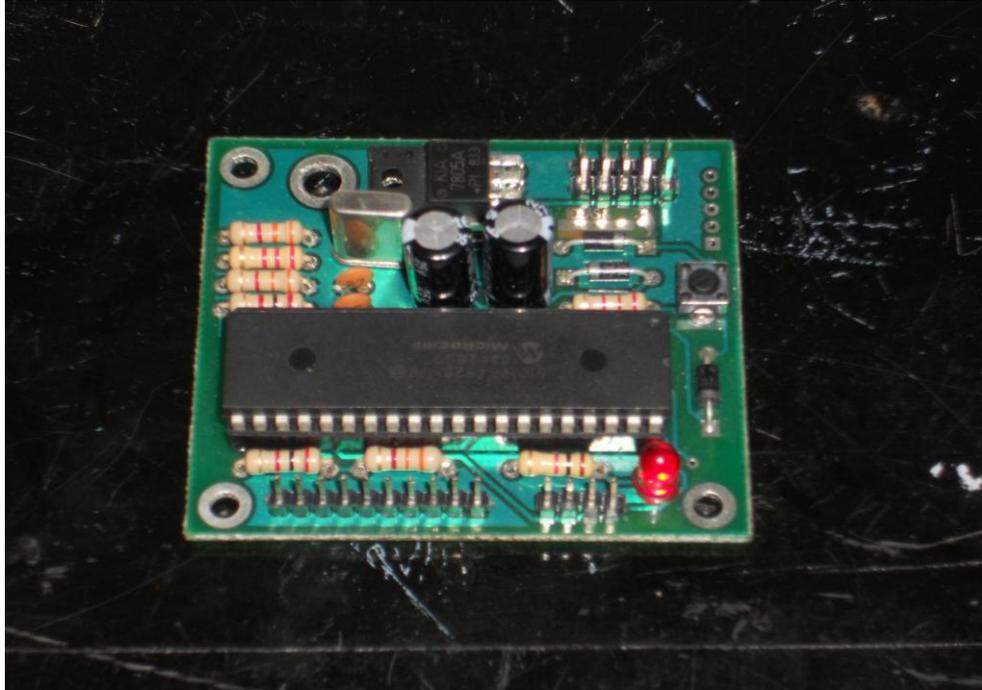
Esquemático del circuito



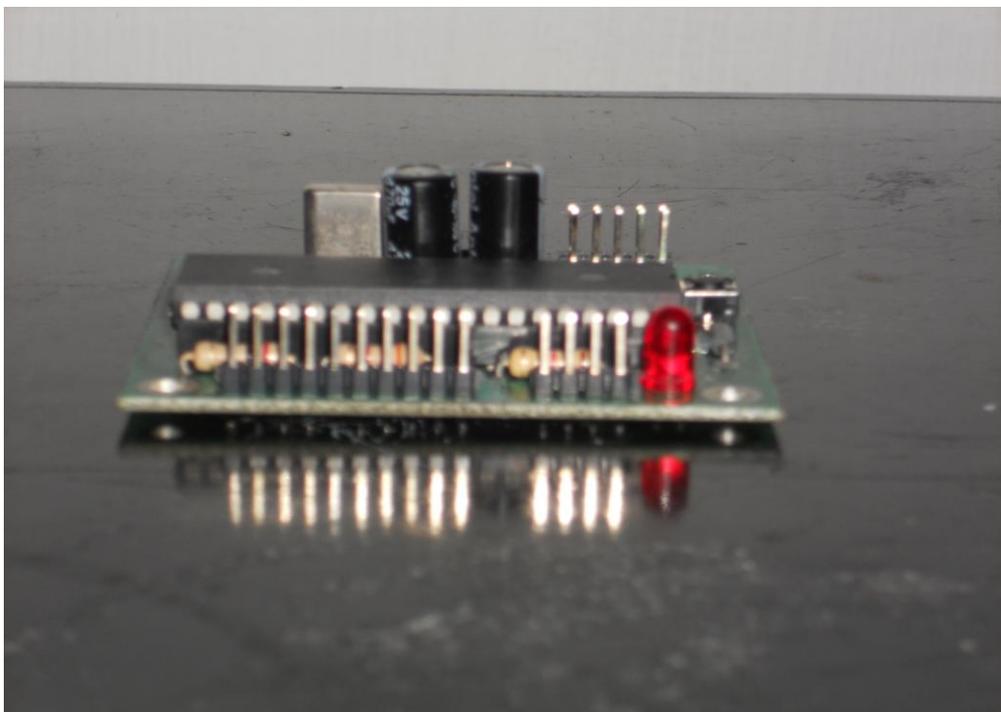
ANEXO 5

Implementación del circuito principal

Cara Superior

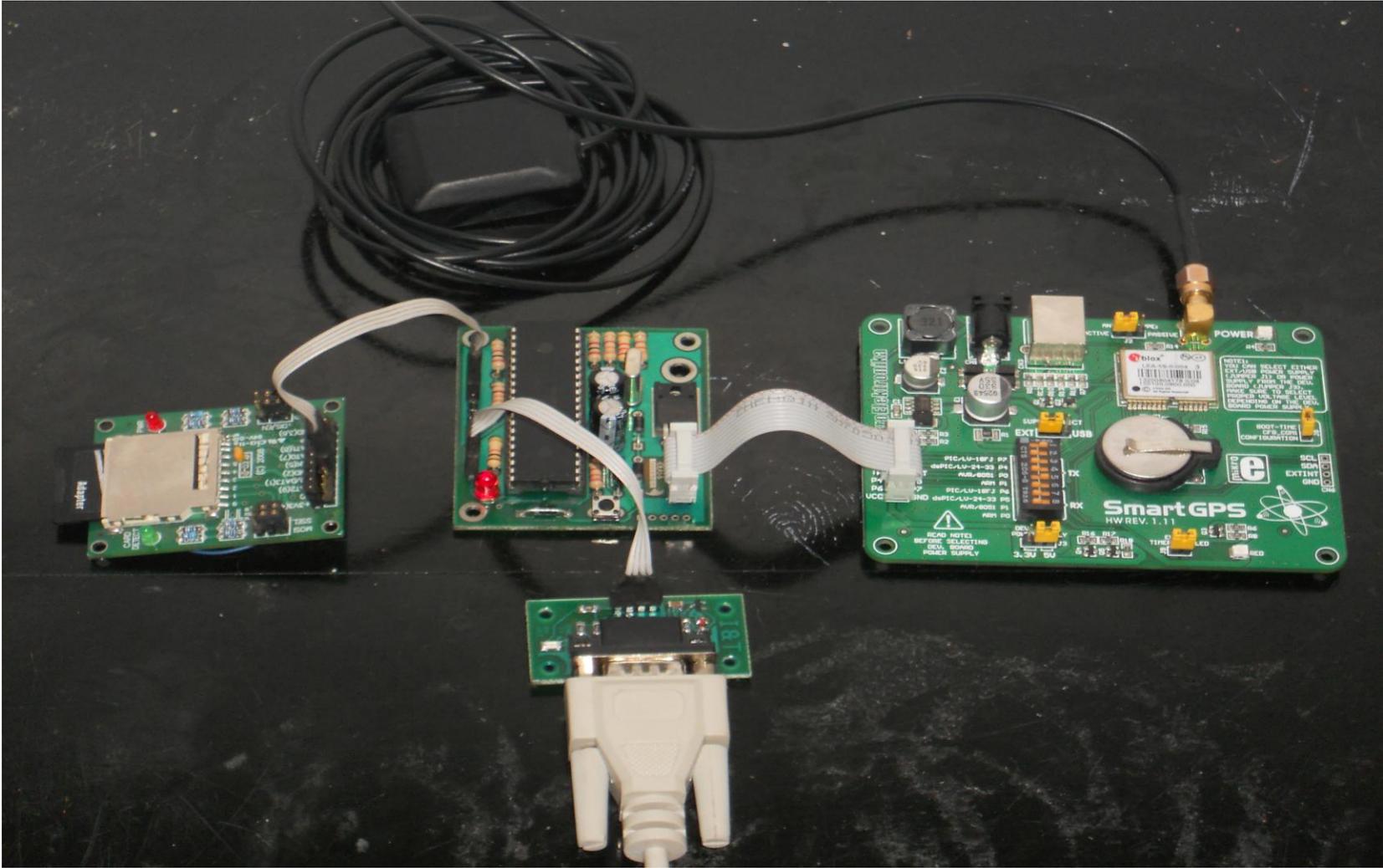


Cara Lateral



ANEXO 6

Ensamblaje final del proyecto



ANEXO 7

Lista de Materiales y Presupuesto

Cant.	Descripción	Precio U	Precio T
1	Tarjeta SmartGPS	\$ 100,00	\$ 100,00
1	Tarjeta ET-MINI SD/MMC	\$ 50,00	\$ 50,00
1	PIC18F4520 microcontrolador	\$ 10,00	\$ 10,00
1	Sandisk Tarjeta de memoria microSD 1Gb	\$ 12,00	\$ 12,00
8	Resistores 1/2W	\$ 0,03	\$ 0,24
1	Oscilador de cuarzo de 20Mhz	\$ 1,50	\$ 1,50
3	Diodo 1N4004	\$ 0,10	\$ 0,30
2	Capacitores electrolíticos 50V	\$ 0,25	\$ 0,50
2	Capacitores cerámicos	\$ 0,10	\$ 0,20
1	LM7805 Regulador de voltaje +5V	\$ 0,25	\$ 0,25
1	Tira de espadines macho	\$ 0,40	\$ 0,40
1	Tira de espadines hembra	\$ 1,20	\$ 1,20
1	Diodo LED 5mm	\$ 0,10	\$ 0,10
1	Botonera de dos pines	\$ 0,10	\$ 0,10
1	Max232	\$ 3,00	\$ 3,00
1	Conector DB9 hembra	\$ 0,60	\$ 0,60
1	Zócalo de 40 pines	\$ 0,40	\$ 0,40
1	Zócalo de 18 pines	\$ 0,20	\$ 0,20
1	Fabricación PCB	\$ 50,00	\$50,00
		TOTAL	\$ 230,99

BIBLIOGRAFÍA

- [1]. Microchip, Data Sheet PIC18F4520;
<http://www.datasheetcatalog.org/datasheet2/f/0xwypjaz882icwuhol9cl7exy37y.pdf> ; **Fecha de consulta:** 20/Junio/2010.
- [2]. U-blox; Data Sheet LEA-5X;
<http://www.u-blox.com/en/download-center.html> ; **Fecha de consulta:** 20/Julio/2010.
- [3]. Mikroelektronika, Diagrama esquemático de la tarjeta Smart GPS v1.01;
http://www.mikroe.com/pdf/smartgps_schematic_v101.pdf ; Fecha de consulta: 04/Agosto/2010.
- [4]. Mikroelektronika; Presentación Tarjeta Smart GPS;
http://www.mikroe.com/eng/downloads/get/37/mikrobasic_pic_pro_manual_v101.pdf ; **Fecha de consulta:** 04/Agosto/2010.
- [5]. Mikroelektronika; Manual de Usuario dsPICPRO4;
http://www.mikroe.com/eng/downloads/get/81/dspicpro4_manual_v101.pdf ; **Fecha de consulta:** 05/Agosto/2010.
- [6]. Mikroelektronika; Guía de Referencia MikroC;
http://www.mikroe.com/eng/downloads/get/945/es_c_syntax_v102.pdf ;
Fecha de consulta: 12/Agosto/2010.
- [7]. Mikroelektronika; Manual de Usuario MikroBasic Pro for PIC;
http://www.mikroe.com/eng/downloads/get/37/mikrobasic_pic_pro_manual_v101.pdf ; **Fecha de consulta:** 12/Agosto/2010.
- [8]. Mikroelektronika; Guía de Referencia MikroBasic;
http://www.mikroe.com/pdf/mikrobasic/basic_syntax_v101.pdf ; **Fecha de consulta:** 14/Agosto/2010.

- [9]. Ingeniería en Microcontroladores; Protocolo SPI (Serial Peripheral Interface); <http://www.i-micro.com/pdf/articulos/spi.pdf> ;
Fecha de consulta: 15/Agosto/2010.
- [10]. Mikroelektronika; GPS data logger with SD card storage;
http://www.mikroe.com/eng/downloads/get/792/gps_data_logger_ew_11_09.pdf ; **Fecha de consulta:** 17/Agosto/2010.
- [11]. ETT; Manual de Usuario ET-MINI SD/MMC;
<http://mac6.ma.psu.edu/space2008/RockSat/microController/MINISD.pdf> ;
Fecha de consulta: 18/Agosto/2010.
- [12]. Arquimedes; NMEA 0183 COMUNICACION PROTOCOL;
<http://www.arquimedes.com.mx/guides/es/mu/NMEA%200183%20protocolo%20v0508es.pdf> ; **Fecha de consulta:** 28/Agosto/2010.