



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

“Aplicación de Módulo Receptor de GPS para el posicionamiento de robots
manejados a control remoto con capacidad de comunicación serial a
Datalogger e Interfaz Gráfica”

TESINA DE SEMINARIO

PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

**INGENIERO EN ELECTRICIDAD ESPECIALIZACIÓN ELECTRÓNICA Y
AUTOMATIZACIÓN INDUSTRIAL**

**INGENIERO EN COMPUTACIÓN ESPECIALIZACIÓN SISTEMAS
TECNOLÓGICOS**

PRESENTADO POR:

Martin Luther Avilés Avilés

Juan Carlos Bajaña Gutiérrez

GUAYAQUIL – ECUADOR

2010

AGRADECIMIENTO

A Dios por guiar mi camino.

A la ESPOL y mis profesores por los conocimientos recibidos.

A mis compañeros y amigos por su apoyo incondicional.

Martin Avilés

DEDICATORIA

Le dedico el presente trabajo a mi esposa, a mis padres y a mis tíos, quienes no solo han sido el pilar en que me apoyo siempre sino también la fuente de inspiración y ganas para seguir adelante.

Martin Avilés

AGRADECIMIENTO

Ante todo a Dios por permitir que culmine esta etapa de mi vida, luego a mis padres por darme la oportunidad de una excelente educación y estar siempre a mi lado. A mi hermana que siempre me brindó su apoyo. Al Ing. Carlos Valdivieso por compartir sus conocimientos y estar siempre dispuesto a ayudar.

Juan Carlos Bajaña

DEDICATORIA

A mis padres que me han demostrado que todo se puede lograr con trabajo duro y perseverancia. A mis abuelos que me están guiando desde el cielo.

Juan Carlos Bajaña

TRIBUNAL DE SUSTENTACIÓN



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

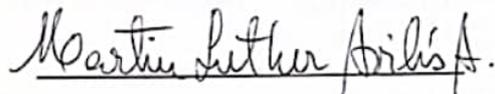


Ing. Hugo Villavicencio V.
DELEGADO DEL DECANO

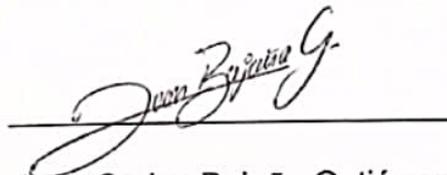
DECLARACIÓN EXPRESA

"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).



Martin Luther Avilés Avilés



Juan Carlos Bajiña Gutiérrez

RESUMEN

El Sistema de Posicionamiento Global GPS es actualmente una de las principales tecnologías utilizadas para fines de orientación. Sus aplicaciones se pueden notar en sistemas de posicionamiento de robots móviles, navegación para vehículos, ubicación para exploradores y deportistas.

En este trabajo se describe la conexión entre un receptor GPS y un microcontrolador, la determinación de las coordenadas en latitud y longitud, así como también su visualización en un mapa ajustado, que pueden ser utilizados en un sistema de posicionamiento para robots.

Se aprovecha la tarjeta de desarrollo SmartGPS como una herramienta educativa, además de proveer una comunicación sencilla entre el microcontrolador y el módulo GPS, para el desarrollo de habilidades en el uso de esta nueva tecnología en este medio.

ABREVIATURAS

GLCD Graphic Liquid Crystal Display (Pantalla de cristal líquido)

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)

FIEC Facultad de Ingeniería de Electricidad y Computación

PuTTY (Port Unique Terminal Type) Puerto Unico Tipo Terminal

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 Recomendend Standard 232 (Estándar Recomendado 232)

SPI (Serial Peripheral Interface) Interfaz Perfiérica Serial

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

INDICE GENERAL

INTRODUCCIÓN

1	Descripción General del Sistema	1
1.1	Alcance y Limitaciones del Proyecto.....	15
1.1.1	Descripción del Proyecto.....	15
1.1.2	Estrategia Implementada	2
1.1.3	Limitaciones del Proyecto	2
1.2	Análisis de Soluciones existentes en el mercado	3
2	Fundamentación Teórica.....	5
2.1	Que es un GPS y como funciona.....	5
2.2	El protocolo NMEA.....	6
2.3	Comunicación serial.....	8
2.3.1	Tipos de transmisión serial.....	9
2.4	Conceptos Básicos de Programación	10
2.5	Herramientas de software	11
2.5.1	MikroBasic Pro for PIC.....	11
2.5.2	Proteus.....	11
2.5.3	HyperTerminal.....	12
2.5.4	PuTTY	12
2.5.5	Virtual Serial Port Driver.....	12
2.6	Principales características del PIC18F452 – I/P	13
3	Diseño del proyecto.....	14
3.1	Diseño general.....	14
3.2	Diseño de la aplicación	15
3.2.1	Especificaciones funcionales	15
4	Simulación y Pruebas Experimentales	62
4.1	Verificación de la Transmisión de Datos del módulo GPS	62
4.1.1	HyperTerminal.....	63
4.1.2	Programa PUTTY.....	63

4.1.3	Terminal Virtual del Proteus	64
4.2	Simulaciones Realizadas	65
4.2.1	Ejemplo Mapamundi en Proteus	65
4.2.2	Proyecto en Proteus.....	67

CONCLUSIONES

RECOMENDACIONES

ANEXOS

BIBLIOGRAFÍA

INDICE DE FIGURAS

Figura 3.1 Diagrama de bloques del proyecto	15
Figura 3.2 El microcontrolador	57
Figura 3.3 Tarjeta de desarrollo SmartGPS	59
Figura 3.4 Pantalla Gráfica GLCD 128X64	60
Figura 3.4 Configuración del MAX232	61
Figura 4.1 Prueba con el HyperTerminal	63
Figura 4.2 Prueba con el programa PuTTY	64
Figura 4.3 Prueba realizada con Proteus usando el Virtual Terminal	65
Figura 4.4 Simulación con Proteus	66
Figura 4.5 Simulación en ejecución con Proteus	67
Figura 4.6 Croquis de la planta alta del edificio principal FIEC.....	68
Figura 4.7 Croquis ya convertido a Bitmap	68
Figura 4.8 Simulación utilizando croquis en Bitmap.....	73

INDICE DE TABLAS

Tabla 1.1 Modelos de tarjetas GPS	4
Tabla de 3.1 Definición de variables para el uso librerías GLCD.....	19

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, es posible determinar las coordenadas de la ubicación del robot. Información que puede ser procesada, a través de un microcontrolador y así poder mostrar dicha ubicación mediante una interfaz gráfica.

En el Capítulo 1 se describe en detalle el sistema para conocer su alcance y limitaciones, así como también la estrategia implementada. En este capítulo se analizan también otros productos similares en el mercado.

En el Capítulo 2 se examina la fundamentación teórica referente a cada uno de los dispositivos y elementos que conforman el sistema.

En el Capítulo 3 se detalla el diseño general, empezando con la implementación física y pasando luego a la programación de los dispositivos.

Luego en el Capítulo 4 se presentan las simulaciones realizadas, detalles de la implementación y datos obtenidos en diferentes pruebas.

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 y una GLCD. Mediante este sistema se puede determinar y ajustar las coordenadas en un mapa a escala del edificio de la Facultad de Electricidad y Computación.

1.1 Alcance y Limitaciones del Proyecto

Se analiza a continuación 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 brindar una determinada ubicación en las inmediaciones del edificio principal de la FIEC mediante un sistema de coordenadas proporcionado por un módulo GPS. Esta ubicación podrá ser visualizada en un mapa a escala, mediante una pantalla gráfica LCD.

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 USART del microcontrolador, un PIC18F452. Mediante este microcontrolador 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, ordenará estos datos y los procesará, finalmente entregará la información de la ubicación a través de la GLCD, mostrando en un mapa a escala la ubicación dentro del edificio principal de la FIEC.

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.

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

CAPITULO 2

2 Fundamentación Teórica

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

2.1 Que es un GPS y como funciona.

El GPS es un sistema de posicionamiento global que permite determinar en todo el mundo la posición de un objeto. 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 éste se encuentra. 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.

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 determinados, el receptor GPS mantendrá en su salida el mismo juego de cadenas, dejando de lado cualquier dato perdido. 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

A continuación se muestra un ejemplo de una cadena NMEA completa:

\$GPGLL,4916.45,N,12311.12,W,225444,A,A,*64

La cadena \$GPGLL provee de datos acerca de la longitud, latitud y del tiempo. La interpretación de este ejemplo es como sigue:

4916.46,N Latitud 49 grados 16.45 min. Norte

12311.12,W Longitud 123 grados 11.12 min. Oeste

225444 Lectura tomada a las 22:54:44 del Tiempo Universal

Coordinado

A Dato válido

A Fix flag

*64 Checksum

2.3 Comunicación serial

Una de las mayores ventajas de esta aplicación fue la de contar con una comunicación serial, utilizando el estándar RS232.

En la comunicación serie los bits se transmiten uno detrás de otro (de ahí el nombre), lo que hace que sean mucho más lentas que sus homólogas "paralelo" en las 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.

2.3.1 Tipos de transmisión serial

Existen varias formas de transmisiones serie:

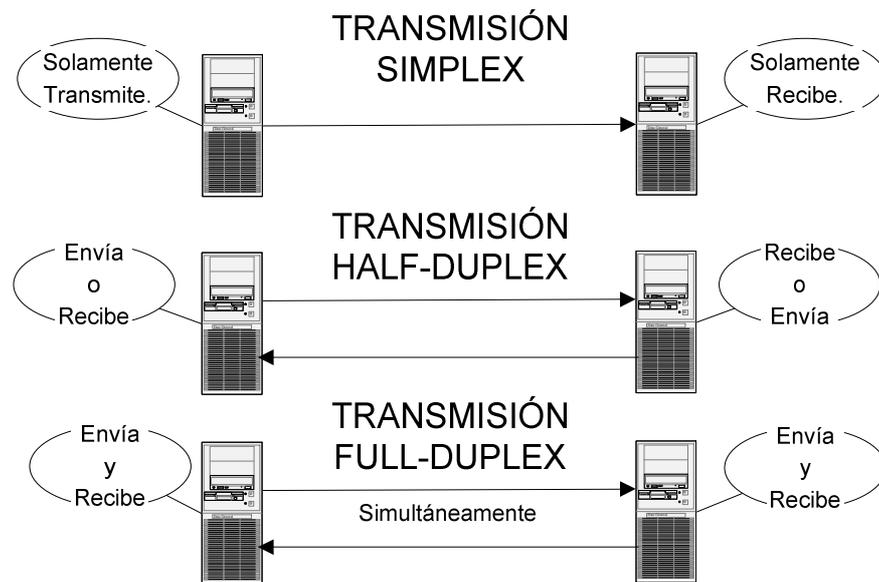


Figura 2.1 Tipos de comunicación serial

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 2.1.

2.4 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.

2.5 Herramientas de software

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

2.5.1 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.

2.5.2 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.

2.5.3 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.

2.5.4 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.

2.5.5 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.

2.6 Principales características del PIC18F452 – I/P

Este microcontrolador pertenece a la familia de microcontroladores de 8-bits del tipo flash. Su capacidad de memoria de programa es de 32kB y su memoria RAM 1,536 Bytes.

Soporta los protocolos de comunicación EUSART (USART mejorado, compatible con los estándares RS232 y RS485), SPI e I2C.

Tiene dos módulos PWM, oscilador interno de hasta 8MHz y convertidor A/D de 8 canales.

De los 40 pines que posee, 35 pueden ser usados como pines de entradas o salidas.

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

Este proyecto tiene como propósito la implementación de un módulo GPS para determinar la ubicación de un robot, con la mayor precisión posible, en las inmediaciones del edificio principal de la FIEC. Para este fin se ha considerado un mapa ajustado de estas instalaciones.

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 \$GPGLL. Esta cadena contiene la información necesaria para que el microcontrolador

calcule la ubicación presentando un punto en el mapa ajustado. Véase Figura 3.1.

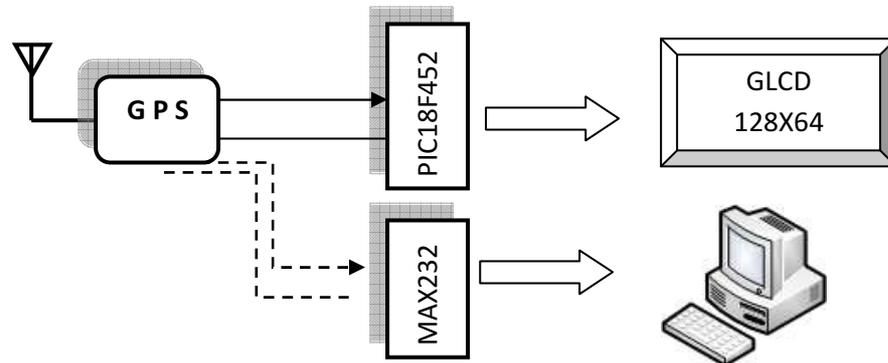


Figura 3.1 Diagrama de bloques del proyecto

3.2 Diseño de la aplicación

3.2.1 Especificaciones funcionales

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

3.2.1.1 Diseño de los componentes de software

3.2.1.1.1 Código SmartGPS

Para programar el PIC 18F452 se usa la herramienta de software **MikroBasic Pro for PIC** de Mikroelectrónica, el primer bloque corresponde a la declaración de variables, constantes y definición de variables que se utilizan para el manejo de la pantalla GLCD.

El segundo bloque está dedicado a las funciones y procedimientos que se van a utilizar en el programa principal, estas funciones o procedimientos pueden ser llamadas varias veces en el programa principal.

El tercer bloque es el bloque principal, en este bloque se encuentra la inicialización de variables, GLCD, UART1 y pines del PIC. Luego se tiene la sección de código que recibe los datos del módulo GPS y que muestra los datos en la pantalla GLCD.

3.2.1.1.2 Declaración de variables

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

```
dim txt as char[768]
```

```
str_ as string[40]
```

```
tmpStr as string[10]
```

```
latitude, longitude as float
```

```
nmbByte, i, g, cnt as integer
```

```
ready as byte
```

El código dentro del recuadro azul corresponde a la declaración de variables globales que se van a usar para la captura de datos del

módulo GPS, procesamiento de estos datos para mostrarlos en la pantalla GLCD, además de una bandera que me indica cuando esta listo para recibir el siguiente dato a mostrar.

```
dim GLCD_Dataport as byte at PORTD
```

```
dim GLCD_CS1 as sbit at RB0_bit
```

```
GLCD_CS1_Direction as sbit at TRISB0_bit
```

```
GLCD_CS2 as sbit at RB1_bit
```

```
GLCD_CS2_Direction as sbit at TRISB1_bit
```

```
GLCD_RS as sbit at RB2_bit
```

```
GLCD_RS_Direction as sbit at TRISB2_bit
```

```
GLCD_RW as sbit at RB3_bit
```

```
GLCD_RW_Direction as sbit at TRISB3_bit
```

```
GLCD_EN as sbit at RB4_bit
```

```
GLCD_EN_Direction as sbit at TRISB4_bit
```

```
GLCD_RST as sbit at RB5_bit
```

```
GLCD_RST_Direction as sbit at TRISB5_bit
```

Este código corresponde a la definición de las variables que se usan para el manejo de la pantalla GLCD, esta definición es muy importante, ya que en esta se define cual es el puerto de datos para la pantalla GLCD, además de las dependencias externas de la librería Gráfica LCD, estas se detallan en la Tabla 3.1'

Touch Panel module connections

```
dim DRIVE_A as sbit at RB6_bit ' for writing to output pin always use  
latch (PIC18 family)
```

```
    DRIVE_B as sbit at RB7_bit ' for writing to output pin always use  
latch (PIC18 family)
```

```
    DRIVE_A_Direction as sbit at TRISB6_bit
```

```
    DRIVE_B_Direction as sbit at TRISB7_bit
```

```
const READ_X_CHANNEL = 0 ' READ X line is connected to analog  
channel 0
```

```
const READ_Y_CHANNEL = 1 ' READ Y line is connected to analog  
channel 1
```

```
' End Touch Panel module connections
```

Este código corresponde a la declaración de variables para el manejo del touch panel, se define al puerto RB6 como DRIVE_A y al puerto RB7 como DRIVE_B.

The following variables must be defined in all projects using Graphic Lcd Library:	Description:
<code>dim GLCD_DataPort as byte sfr external</code>	Glcd Data Port.
<code>dim GLCD_CS1 as sbit sfr external</code>	Chip Select 1 line.
<code>dim GLCD_CS2 as sbit sfr external</code>	Chip Select 2 line.
<code>dim GLCD_RS as sbit sfr external</code>	Register select line.
<code>dim GLCD_RW as sbit sfr external</code>	Read/Write line.
<code>dim GLCD_RST as sbit sfr external</code>	Reset line.
<code>dim GLCD_EN as sbit sfr external</code>	Enable line.
<code>dim GLCD_CS1_Direction as sbit sfr external</code>	Direction of the Chip Select 1 pin.
<code>dim GLCD_CS2_Direction as sbit sfr external</code>	Direction of the Chip Select 2 pin.
<code>dim GLCD_RS_Direction as sbit sfr external</code>	Direction of the Register select pin.
<code>dim GLCD_RW_Direction as sbit sfr external</code>	Direction of the Read/Write pin.
<code>dim GLCD_EN_Direction as sbit sfr external</code>	Direction of the Enable pin.
<code>dim GLCD_RST_Direction as sbit sfr external</code>	Direction of the Reset pin.

Tabla de 3.1 Definición de variables para el uso librerías GLCD

```
dim x_coord, y_coord, x_coord_old, y_coord_old, x_coord_diff,
y_coord_diff as word
```

```
x_coord128, y_coord64 as longint ' scaled x-y position
```

```
cal_x_min, cal_y_min, cal_x_max, cal_y_max as word ' calibration
constants
```

```
menu as byte ' flag menu
```

```
texto as string[20] ' para presentar los datos
```

```
flag1 as byte
```

```
flag2 as byte
```

```
const ADC_THRESHOLD as word = 900 ' Threshold
value for press detecting
```

En este bloque de código se declaran las variables para la conversión de las coordenadas en X y Y que indican el punto donde se está presionando el touch panel. Además se declaran las variables que funcionan como banderas para la presentación del menú.

3.2.1.1.2.1 Funciones y Procedimientos

3.2.1.1.2.1.1 Funciones

En este programa se creó función **absolut** cuyo objetivo es convertir valores negativos a positivos.

```
sub function absolut (dim a as integer ) as integer

    result = a

    if (a < 0) then

        result = 0 - a

    end if

end sub
```

La función **PressDetect** retorna 1 cuando se ha presionado el touch panel y 0 en el caso contrario.

```
sub function PressDetect() as byte

    dim adc_rd as word

    ' PRESS detecting

    DRIVE_A = 0    ' DRIVEA = 0 (LEFT drive off, RIGHT drive off, TOP
drive on)

    DRIVE_B = 0    ' DRIVEB = 0 (BOTTOM drive off)
```

```
Delay_ms(5)
```

```
adc_rd = ADC_Read(READ_Y_CHANNEL) ' READ-Y
```

```
result = (adc_rd > ADC_THRESHOLD) ' if logical one is detectet
```

```
'debouncing, repeat detecting after 2ms
```

```
Delay_ms(2)
```

```
adc_rd = ADC_Read(READ_Y_CHANNEL) ' READ-Y
```

```
result = result and (adc_rd > ADC_THRESHOLD)
```

```
end sub
```

La función **GetX** obtiene la coordenada X del punto que se presiona en el touch panel.

```
sub function GetX() as word
```

```
'reading X
```

```
DRIVE_A = 1 ' DRIVEA = 1 (LEFT drive on, RIGHT drive on,  
TOP drive off)
```

```
DRIVE_B = 0      ' DRIVEB = 0 (BOTTOM drive off)
```

```
Delay_ms(5)
```

```
result = ADC_Read(READ_X_CHANNEL)  ' READ-X (BOTTOM)
```

```
end sub
```

La función **GetY** obtiene la coordenada X del punto que se presiona en el touch panel.

```
sub function GetY() as word
```

```
'reading Y
```

```
DRIVE_A = 0      ' DRIVEA = 0 (LEFT drive off, RIGHT drive off,  
TOP drive on)
```

```
DRIVE_B = 1      ' DRIVEB = 1 (BOTTOM drive on)
```

```
Delay_ms(5)
```

```
result = ADC_Read(READ_Y_CHANNEL)  ' READ-X (LEFT)
```

```
end sub
```

La función **search_str2_in_str1** para obtener la línea de texto que envía el módulo GPS que corresponde a la posición geográfica en longitud, latitud y el tiempo, esta función realiza comparaciones de los caracteres del string 2 en el string 1. El código de la función se detalla a continuación:

```
sub function search_str2_in_str1(dim byref s1 as string[4000], dim byref  
s2 as string[4000]) as word
```

```
    dim i, j as word
```

```
        aa, bb as byte
```

```
        i = 0
```

```
        j = 0
```

```
        aa = s1[0]
```

```
        bb = s2[0]
```

```
        result = 0xFFFF
```

```
        while(aa <> 0)
```

```
            while(aa = bb)
```

```
                if (i = 0) then
```

```
    result = j

end if

i = i + 1

j = j + 1

aa = s1[j]

bb = s2[i]

if (bb = 0) then

    exit

end if

wend

i = 0

j = j + 1

aa = s1[j]

bb = s2[i]

result = 0xFFFF

wend
```

```
end sub
```

3.2.1.1.2.1.2 Procedimientos

El procedimiento **Calibrate** solicita al usuario que determine los puntos inferior izquierdo y superior derecho para establecer los puntos máximos y mínimos del touch panel, estos valores son importantes para la conversión de las coordenadas X y Y a 128 y 64 pixeles respectivamente.

```
sub procedure Calibrate()
```

```
    Glcd_Dot(0,63,1)
```

```
    Glcd_Write_Text("TOUCH BOTTOM LEFT",12,3,1)
```

```
    while ( PressDetect() = 0)
```

```
        nop
```

```
    wend
```

```
    ' get calibration constants (reading and compensating TouchPanel  
    nonlinearity)
```

```
    cal_x_min = GetX() - 10
```

```
cal_y_min = GetY() - 10
```

```
Delay_ms(1000)
```

```
Glcd_Fill(0)
```

```
Glcd_Dot(127,0,1)
```

```
Glcd_Write_Text("TOUCH UPPER RIGHT",12,4,1)
```

```
while ( PressDetect() = 0)
```

```
    nop
```

```
wend
```

```
' get calibration constants (reading and compensating TouchPanel  
nonlinearity)
```

```
cal_x_max = GetX() + 5
```

```
cal_y_max = GetY() + 5
```

```
Delay_ms(1000)
```

```
end sub
```

El procedimiento **Initialize** inicializa los puertos del microcontrolador.

```
sub procedure Initialize()
```

```
    ADCON0 = 0
```

```
    ADCON1 = 0x00
```

```
    TRISA0_bit = 1      ' inputs
```

```
    TRISA1_bit = 1
```

```
    DRIVE_A_Direction = 0  ' Set DRIVE_A pin as output
```

```
    DRIVE_B_Direction = 0  ' Set DRIVE_B pin as output
```

```
    Glcd_Init()           ' Initialize Glcd
```

```
end sub
```

El procedimiento **Display_Cursor** toma los datos de la longitud y latitud que devuelve el módulo GPS y los presenta en la pantalla GLCD, en esta función se definen los puntos mínimos y máximos para “x” y “y”, estos valores se determinaron en la fase experimental, estos valores son específicos para el edificio nuevo de la FIEC.

```
sub procedure Display_Cursor(dim x as float, dim y as float)
```

```
dim latitude_y, longitude_x as integer
```

```
dim latitude_xmin, latitude_xmax, latitude_ymin, latitude_ymax as float
```

```
latitude_xmin = 2.14448683
```

```
latitude_xmax = 2.1448885
```

```
latitude_ymin = 79.9674985
```

```
latitude_ymax = 79.967897
```

```
latitude_y = float((61*(y - latitude_ymin))/(latitude_ymax -  
latitude_ymin)) + 1
```

```
longitude_x = float((125*(x - latitude_xmin))/(latitude_xmax -  
latitude_xmin)) + 1
```

```
if longitude_x > 125 then
```

```
    longitude_x = 125
```

```
end if
```

```
if latitude_y > 61 then
```

```
    latitude_y = 61
```

```
end if
```

```
GLCD_Dot(longitude_x,latitude_y,2)
```

```

GLCD_Dot(longitude_x-1,latitude_y,2) ' Centar, Left, Right dot

GLCD_Dot(longitude_x,latitude_y-1,2)

GLCD_Dot(longitude_x+1,latitude_y,2) ' Right, Upper dot

GLCD_Dot(longitude_x,latitude_y+1,2)

Delay_ms(500)

GLCD_Image( @FIEC3_bmp )' Lower dot, display World map

end sub

```

El procedimiento **interrupt** es implícitamente declarado por **MikroBasic PRO for PIC**, se lo ha redefinido para cuando ocurra una interrupción por el Timer 1, que es el timer que se ha habilitado para el manejo de interrupciones. Cuando ocurre una interrupción por el RCIF se realiza la lectura de los datos enviados por el módulo GPS mediante el UART1.

```

sub procedure interrupt()

if (PIR1.0=1) then ' si la interrupción es generada por TMR1IF

T1CON.0 = 0 ' Set Timer1 on

ready = 1 ' Set data ready

i= 0 ' Reset contador

```

```
    PIR1.0 = 0 ' Clear TMR1IF

end if

if (PIR1.5 = 1) then          ' si la interrupción es generada por RCIF

    txt[i] = UART1_Read()

    Inc(i)

    if (i = 768) then

        i= 0

    end if

    T1CON.0=0

    TMR1L=0xB0

    TMR1H=0x3C

    T1CON.0=1

    PIR1.5=0' Stop Timer 1, Timer1 cuenta desde 15536, Start timer 1,
Set RCIF to 0

end if

end sub
```

3.2.1.1.2.2 Bloque Principal (Main)

Este bloque consta a su vez de dos sub-bloques, el primero es para la inicialización de variables, configuración de pines, configuración del Timer 1, inicialización de la pantalla GLCD e inicialización del puerto UART1. El segundo bloque es para la presentación inicial del proyecto y el menú con las dos opciones disponibles. El tercer bloque comienza con la sentencia while, primero determinamos la opción que ha elegido el usuario presionando la pantalla, si el usuario presiona el botón MAPA se asigna 1 a la bandera menú y si el usuario presiona el botón DATOS se asigna 2 a la bandera menú. Luego se verifica que el dato ya ha sido obtenido y que la bandera menú sea diferente de 0, luego se obtiene la cadena de caracteres correspondientes a la posición geográfica en longitud, latitud y hora, esta cadena se define con la cadena inicial \$GPGLL, para esto se utiliza la función search_str2_in_str1, luego se obtiene la longitud y latitud de esta cadena realizando el respectivo cálculo, una vez obtenidas la longitud y latitud. Dependiendo de la bandera menú se procede a presentar la ubicación actual en la pantalla GLCD usando el procedimiento Display_Cursor con la opción 1 y Las coordenadas de latitud y longitud con la opción 2.

main:

Initialize()

```
Glcd_Set_Font(@font5x7, 5, 7, 32)

Glcd_Fill(0x00)

Delay_ms(100)

ready = 0

T1CON.5 = 1

T1CON.4 = 1           ' Set Timer1 Prescaler to 1:8

PIE1.0 = 1           ' Enable Timer1 interrupt

TMR1L = 0xB0

TMR1H = 0x3C         ' Timer1 starts counting from 15536

PIR1.0 = 0

UART1_Init(9600)

PIE1.5 = 1           ' Clear Timer1 interrupt flag, Enable Usart
Receiver interrupt

INTCON.7 = 1

INTCON.6 = 1         ' Enable Global interrupt and
Peripheral interrupt
```

```
T1CON.0 = 1           ' Start Timer 1
```

```
Glcd_Fill(0)         ' Clear Glcd
```

```
Glcd_Set_Font(@font5x7, 5, 7, 32)  ' Choose font "font5x7"
```

```
Glcd_Write_Text("CALIBRATION", 30, 2, 1)
```

```
Delay_ms(1500)
```

```
Glcd_Fill(0)
```

```
Calibrate()
```

```
Glcd_Fill(0)
```

```
flag1 = 0
```

```
flag2 = 0
```

```
menu = 0
```

```
Glcd_Image(@tortuga_bmp)
```

```
while (flag1 = 0)
```

```
' after a PRESS is detected read X-Y and convert it to 128x64 space
```

```
x_coord = GetX() - cal_x_min
```

```
y_coord = GetY() - cal_y_min
```

```
' When lifting pen from the touchpanel surface GetX and GetY
readings
```

```
' (after correct PressDetect reading) may be incorrect
```

```
x_coord_diff = absolut(x_coord - x_coord_old)    ' compare with old
values
```

```
y_coord_diff = absolut(y_coord - y_coord_old)
```

```
x_coord_old = x_coord                            ' save old values
```

```
y_coord_old = y_coord
```

```
x_coord128 = longint(x_coord * 128) div (cal_x_max - cal_x_min)
```

```
y_coord64 = (64 -(y_coord *64) div (cal_y_max - cal_y_min))
```

```
if (((x_coord128 >=10) and (x_coord128 <= 100))and ((y_coord64  
>=10) and (y_coord64 <=53))) then
```

```
    flag1 = 1
```

```
    Glcd_Image(@Blanco_bmp)
```

```
else
```

```
    flag1 = 0
```

```
end if
```

```
wend
```

```
Glcd_Image(@Titulo_bmp)
```

```
Delay_ms(500)
```

```
while (flag2 = 0)
```

```
' after a PRESS is detected read X-Y and convert it to 128x64 space
```

```
x_coord = GetX() - cal_x_min
```

```
y_coord = GetY() - cal_y_min
```

' When lifting pen from the touchpanel surface GetX and GetY readings

' (after correct PressDetect reading) may be incorrect

x_coord_diff = absolut(x_coord - x_coord_old) ' compare with old values

y_coord_diff = absolut(y_coord - y_coord_old)

x_coord_old = x_coord ' save old values

y_coord_old = y_coord

x_coord128 = longint(x_coord * 128) div (cal_x_max - cal_x_min)

y_coord64 = (64 -(y_coord *64) div (cal_y_max - cal_y_min))

if (((x_coord128 >=10) and (x_coord128 <= 100))and ((y_coord64 >=10) and (y_coord64 <=53))) then

flag2 = 1

Glcd_Image(@Blanco_bmp)

else

```
        flag2 = 0

    end if

wend

Delay_ms(500)

while TRUE

    RCSTA.1 = 0

    RCSTA.2 = 0

    if menu = 0 then

        Glcd_Image(@MenuGPS_bmp)

    end if

    if (PressDetect() <> 0) then

        ' after a PRESS is detected read X-Y and convert it to 128x64 space

        x_coord = GetX() - cal_x_min
```

```
y_coord = GetY() - cal_y_min
```

' When lifting pen from the touchpanel surface GetX and GetY readings

' (after correct PressDetect reading) may be incorrect

```
x_coord_diff = absolut(x_coord - x_coord_old)    ' compare with old values
```

```
y_coord_diff = absolut(y_coord - y_coord_old)
```

```
x_coord_old = x_coord                            ' save old values
```

```
y_coord_old = y_coord
```

if ((x_coord_diff>50) or (y_coord_diff>50)) then ' if difference is too big then ignore the reading

```
continue
```

```
end if
```

```
x_coord128 = longint(x_coord * 128) div (cal_x_max - cal_x_min)
```

$y_coord64 = (64 - (y_coord * 64) \text{ div } (cal_y_max - cal_y_min))$

if ((x_coord128 < 0) or (x_coord128 > 127)) then

 continue

end if

if ((y_coord64 < 0) or (y_coord64 > 63)) then

 continue

end if

if menu = 0 then

 if (y_coord64 < 39) and (y_coord64 > 18) then

 'if mapa is pressed

 if ((x_coord128 < 59) and (x_coord128 > 6)) then

 menu = 1

 end if

```
'if datos is pressed

if ((x_coord128 < 120) and (x_coord128 > 67)) then

    menu = 2

end if

end if

else

    menu = 0

end if

end if ' if (PressDetect())

if (ready = 1) and (menu <> 0) then          ' if the data in txt array
is ready do:

    ready = 0

    nmbByte = search_str2_in_str1(txt,"$GPGLL")

    cnt = 0

    for g = nmbByte to nmbByte+39
```

```

str_[cnt] = txt[g]

inc(cnt)

next g

if (nmbByte <> 0xFFFF) then

    if (str_[7] <> ",") then

        longitude = (str_[7]-48)*10 + (str_[8]-48) + (((str_[9]-48)*10) +
(str_[10]-48) + ((str_[12]-48)*0.1) + ((str_[13]-48)*0.01) + ((str_[14]-
48)*0.001) + ((str_[15]-48)*0.0001) + ((str_[16]-48)*0.00001)) / 60)

        latitude = (str_[20]-48)*100 + (str_[21]-48)*10 + (str_[22]-48) +
(((str_[23]-48)*10 + (str_[24]-48) + (str_[26]-48)*0.1 + (str_[27]-48)*0.01
+ (str_[28]-48)*0.001 + (str_[29]-48)*0.0001 + (str_[30]-48)*0.00001) /
60)

        end if

    end if

if menu = 1 then

    Glcd_Image(@FIEC3_bmp)

    Display_Cursor(longitude, latitude)

```

else

```
Glcd_Image(@Blanco_bmp)
```

```
texto = "COORDENADAS SMARTGPS"
```

```
Glcd_Write_Text(texto, 5, 1, 1)
```

```
texto = ""
```

```
FloatToStr(longitude, texto)
```

```
texto = "LAT: " + texto
```

```
Glcd_Write_Text(texto, 5, 3, 1)
```

```
texto = ""
```

```
FloatToStr(latitude, texto)
```

```
texto = "LONG: " + texto
```

```
Glcd_Write_Text(texto, 5, 5, 1)
```

end if

```

end if

wend

end.

```

3.2.1.1.3 Bitmap

Este bloque de código contiene el mapa de bits para la imagen del nuevo edificio de la FIEC, que es donde se van a realizar las mediciones del módulo GPS, este mapa de bits se almacena como una constante. Esta información se obtiene utilizando la herramienta **GLCD Bitmap Editor** que viene en el programa **MikroBasic Pro for PIC**.

```

const FIEC3_bmp as byte[1024] = (

    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,255,

    255,255,255,255,255,255,255,255,255,255,255,255,255, 0,

    255,255,255,255,255,255,255,255,255,255,255,255,255,255,

    255,255,255,255,255,255,255,255,255,255,255, 0, 0,255,255,

    255,255,255,255,255,255,255,255,255,255,255,255,255,255,

    255,255,255,255,255,255,255,255,255,255,255,255,255,255,

```

255,255,255,255,254,255, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255, 0,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,255, 0, 0,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,239,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,127,127,

63, 63, 63, 31, 31, 31, 15, 15, 7, 7, 7, 3, 3, 3, 3, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 3, 3,

7, 7, 15, 15, 31, 31, 31, 63, 63,127,127,127,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
252,252,252,252,252,252,252,252,252,252, 4, 4,244,116,244,244,
244,116,244,116,116,244,116,244,244, 4, 7, 1, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0,112,128, 72, 72, 56,228,128,128, 2, 2, 2,132,
132,100, 24, 8, 8,208, 48, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1,
255, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
255,255,255,255,255,255,255,255,255,255,248,248,251,251,251,251,
251,251,251,251,251,251,251,251,251,248,248,248,248,240,192, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 8, 28, 29, 63, 63, 63, 62, 63, 63, 63, 62, 63, 63,
63, 63, 62, 61, 61, 61, 24, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
255,224,240,240,248,248,248,248,248,248,248,248,248,248,248,248,

248,248,248,248,248,248,248,248,248,248,248,248,248,248,248,248,248,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,127,191,191,191,223,239,239,239,247,251,255,
254,254,252,252,252,252,248,240,192, 0, 0,128,128,128,128,128,
128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,
128,128,128,128,128,128,128,128,128,128,128,128,128,128,128, 0,128,
128,128,192,192,224,224,240,240,248,252,252,252,254,254,254,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,127,191,191,191,223,239,239,239,247,247,251,251,
253,253,254,254,254,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255, 0, 0,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255, 0,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,

```

255,255,255,255,255,255,255,255,255,255,255,255,255,255,
253,254,254,254,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255, 0, 0,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255, 0,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,127
)

```

```

const MenuGPS_bmp as byte[1024] = (

```

```

1,254, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3,131,131, 3, 3, 3, 3, 3,131,131, 3, 3,131,131,131,131,
3, 3,131,131, 3, 3, 3,131, 3, 3,131, 3, 3, 3, 3,131,

```


252,252,252,252,252,252,252,252,252,252,252,252,252,252,252,252,
252,252,252,252,252,252,252,252,252,252,252,252,252,252,252,252,
252,252,252,252,252,248,252,240,208,192, 0, 0, 0, 0,255, 0,
0,255, 0, 0, 0, 0,255,255,255,255,255,255,255,255,255,255,
255,255, 3, 3,227,135, 63, 63,143,227, 3, 3,255, 63, 15, 35,
115, 7, 31,255,255, 3, 3,187,147,199,255,127, 31, 3,115, 3,
3, 15,127,255,255,255,255,255,255,255,255,255,255,255, 0, 0,
0, 0,255,255,255,255,255,255,255,255,255,255, 3,251,251,243,
243, 35, 7,255,127, 31, 3,115, 3, 15,127,251,251,251, 3,243,
251,251, 15, 7,243,251,251,115, 7,143,255,195,203,155, 19, 63,
63,255,255,255,255,255,255,255,255,255, 0, 0, 0, 0,255, 0,
0,255, 0, 0, 0, 0, 15, 47, 63,255,255,255,255,255,255,255,
255,255,254,254,255,255,254,254,255,255,254,254,255,254,255,255,
255,255,254,254,255,254,254,255,255,255,255,254,254,255,255,255,
255,254,254,255,255,255,255,255,255,255,255, 63, 31, 31, 0, 0,
0, 0, 15, 63, 63,255,255,255,255,255,255,255,254,254,254,254,

254,254,255,255,254,254,255,255,255,254,254,255,255,255,254,255,
255,255,255,254,254,252,254,254,254,255,254,254,252,254,254,255,
255,255,255,255,255,255,255, 63, 47, 15, 0, 0, 0, 0,255, 0,
0,255, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,255, 0,
0,255, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 64,206,139,217,115, 32, 0,255, 3, 14,120,224,120, 14, 3,
3,255, 0,192,120, 46, 35, 46,120,192, 0,126,255, 17, 27,126,
198, 0, 3, 3,255, 3, 3, 0, 0, 0, 0, 60,126,194,193,217,

6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,250,254, 0,
0,255,255, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0,254,254, 0, 0,254, 6, 12, 56,224,254,
254, 0, 2, 2,254,254, 2, 2, 0,254,254,146,146,146, 0, 0,
0,254,254, 18, 50,238,132, 0, 0,254, 18, 18, 18, 0,128,240,
60, 38, 46,124,240,128,128,226,242,154,142,130, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,255,255, 0,
0,255,255, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1,

1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0,
0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1,
0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 255, 255, 0,
0, 255, 255, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 192, 224, 48, 16, 144, 144, 160, 0,
0, 240, 240, 16, 176, 224, 96, 0, 0, 128, 224, 48, 240, 192, 0, 0,
0, 0, 240, 176, 144, 144, 0, 0, 240, 240, 0, 128, 224, 48, 16, 16,
48, 32, 0, 0, 224, 112, 112, 224, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 255, 255, 0,
0, 255, 255, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 3, 7, 12, 8, 9, 15, 7, 0,

0, 15, 7, 1, 3, 6, 12, 0, 12, 7, 3, 2, 2, 3, 14, 0,
0, 0, 15, 5, 1, 1, 0, 0, 15, 15, 0, 3, 7, 12, 12, 8,
12, 4, 8, 15, 3, 2, 2, 3, 15, 12, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 255, 255, 0,
0, 255, 255, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 128, 128, 128, 128, 0, 0, 128, 128, 0, 0, 0, 0, 0, 128, 128,
0, 0, 0, 0, 128, 128, 128, 0, 0, 0, 0, 128, 128, 128, 128, 0,
0, 0, 128, 128, 128, 128, 128, 128, 128, 0, 0, 0, 0, 0, 128,
128, 128, 128, 128, 0, 0, 128, 128, 128, 128, 0, 0, 0, 0, 128, 128,
128, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 255, 255, 0,
0, 255, 255, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
70, 207, 201, 89, 113, 0, 0, 255, 3, 14, 120, 224, 120, 14, 3, 255,

3.2.1.2 Diseño de los componentes de hardware

A continuación se presentan las características más relevantes de cada uno de los módulos o etapas del proyecto.

3.2.1.2.1 El Microcontrolador

El PIC18F452-I/P es un microcontrolador de 8 bits del tipo flash, con memoria de programa de 32kB y una RAM de 1536 Bytes. Este último dato fue muy importante para la selección del micro ya que la mayoría de los dispositivos disponibles tenían tan solo la mitad de memoria RAM. Además tiene 40 pines, de los cuales 35 pueden ser entradas o salidas.



Figura 3.2 El microcontrolador

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 por I2C (Integer-Integrated Circuit), permitiendo así la comunicación con otros dispositivos compatibles.

3.2.1.2.2 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 3.3.

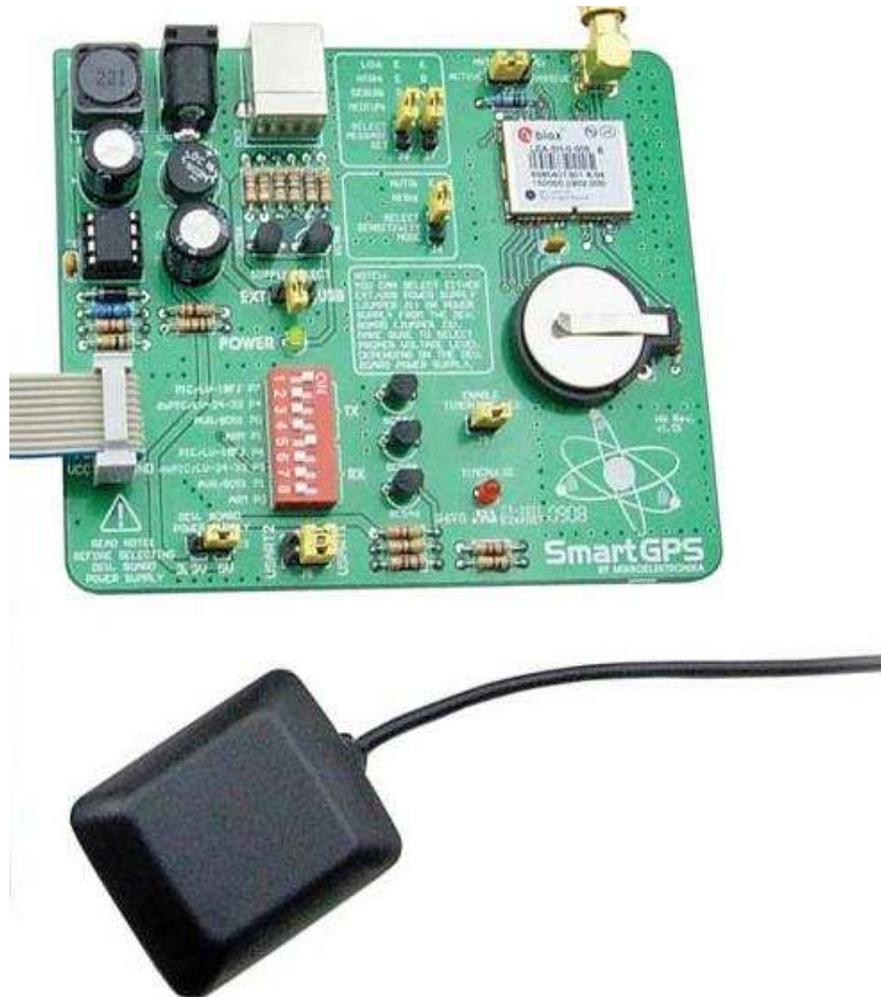


Figura 3.3 Tarjeta de desarrollo SmartGPS

3.2.1.2.3 Pantalla Gráfica GLCD 128X64 con Panel Táctil

La interfaz gráfica es mostrada a través de una pantalla LCD gráfica con resolución de 128X64 pixeles, además cuenta con un panel táctil el cual permite el diseño de aplicaciones con una interacción más amigable con el usuario. La pantalla es monocromática RGB es decir puede mostrar la información en cualquiera de los tonos

que se pueda obtener combinando los colores rojo, verde y azul. Su controlador es un chip Samsung S6B0108 (KS0108).

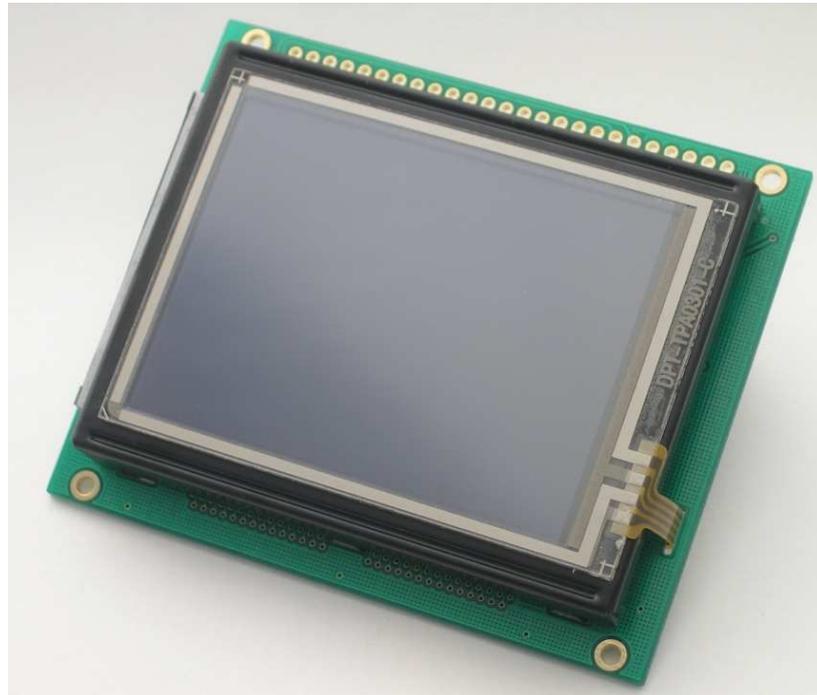


Figura 3.4 Pantalla Gráfica GLCD 128X64

3.2.1.2.4 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 3.4 permitió establecer un vínculo entre la tarjeta GPS y la PC, teniendo así varias alternativas para visualizar las tramas y poder interpretarlas.

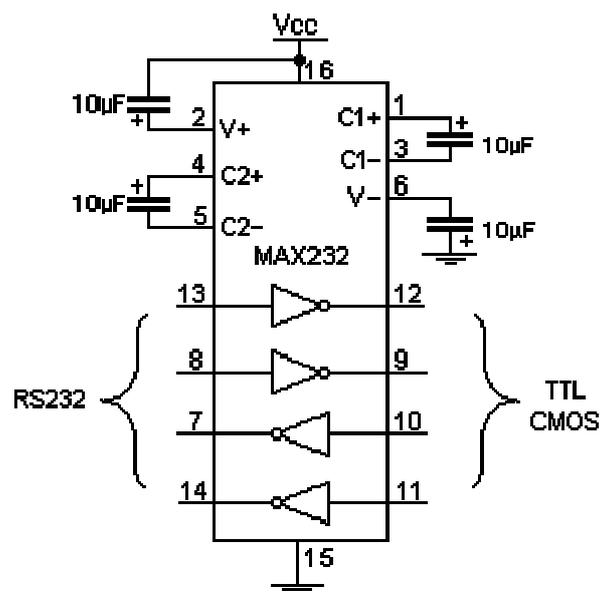


Figura 3.4 Configuración del MAX232

CAPÍTULO 4

4 Simulación y Pruebas Experimentales

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 que viene en Windows XP se procede a configurar el HyperTerminal a una tasa de baudios de 9600 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 sólo se puede realizar con Windows XP.

```

$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

```

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 un 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 y así obtener las lecturas del GPS tal como se muestra en la figura 4.3.

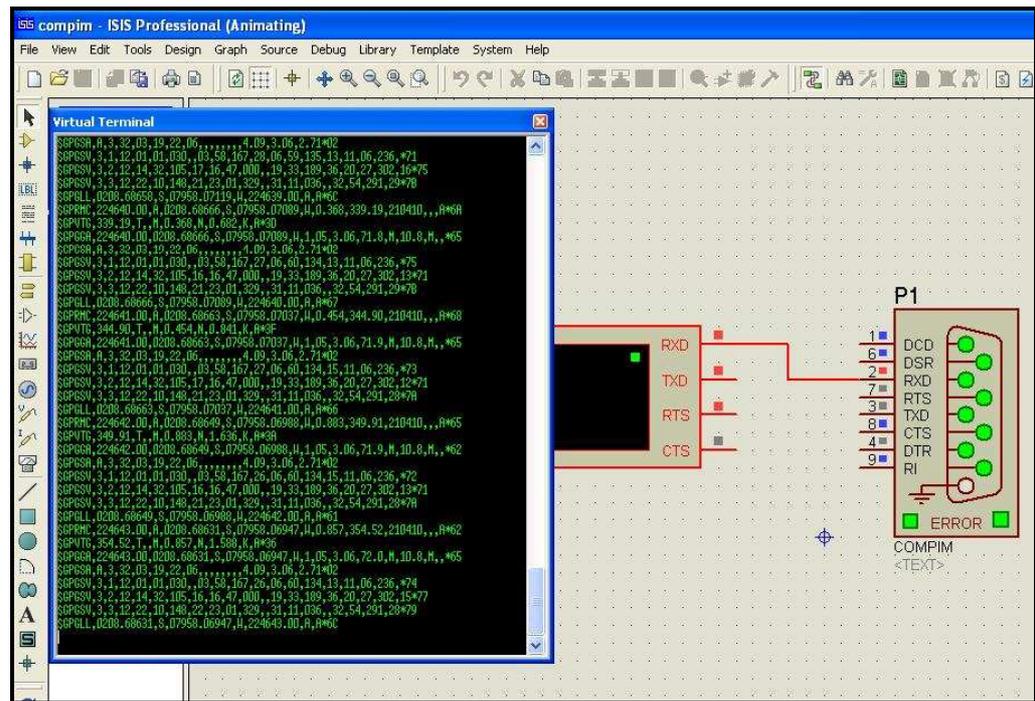


Figura 4.3 Prueba realizada con Proteus usando el Virtual Terminal

4.2 Simulaciones Realizadas

4.2.1 Ejemplo Mapamundi en Proteus

Para la simulación del ejemplo que se encuentra en el artículo [es_mikroe_article_basic_pic_04_09.pdf](#) se realizó la simulación en Proteus del circuito que se muestra en la figura 4.4, luego se creó el proyecto SmartGPS usando el programa **MikroBasic PRO for PIC** tomando las líneas de código que se incluyen en este artículo se

compila el programa y este se lo carga en el PIC 18F4520 de la simulación en Proteus.

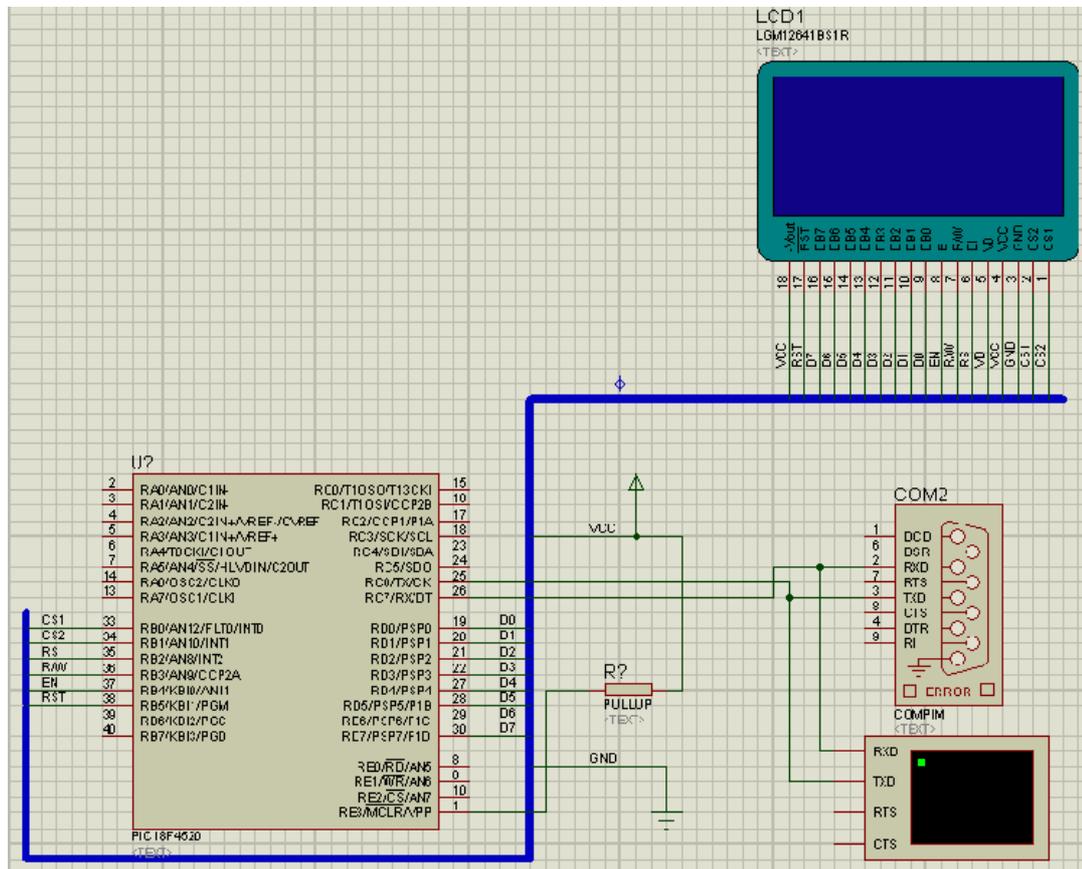


Figura 4.4 Simulación con Proteus

Se configura el COMPIM para configurar el puerto físico donde se conecta 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 y se ejecuta la simulación, el resultado debe presentarse como se muestra en la figura 4.5.

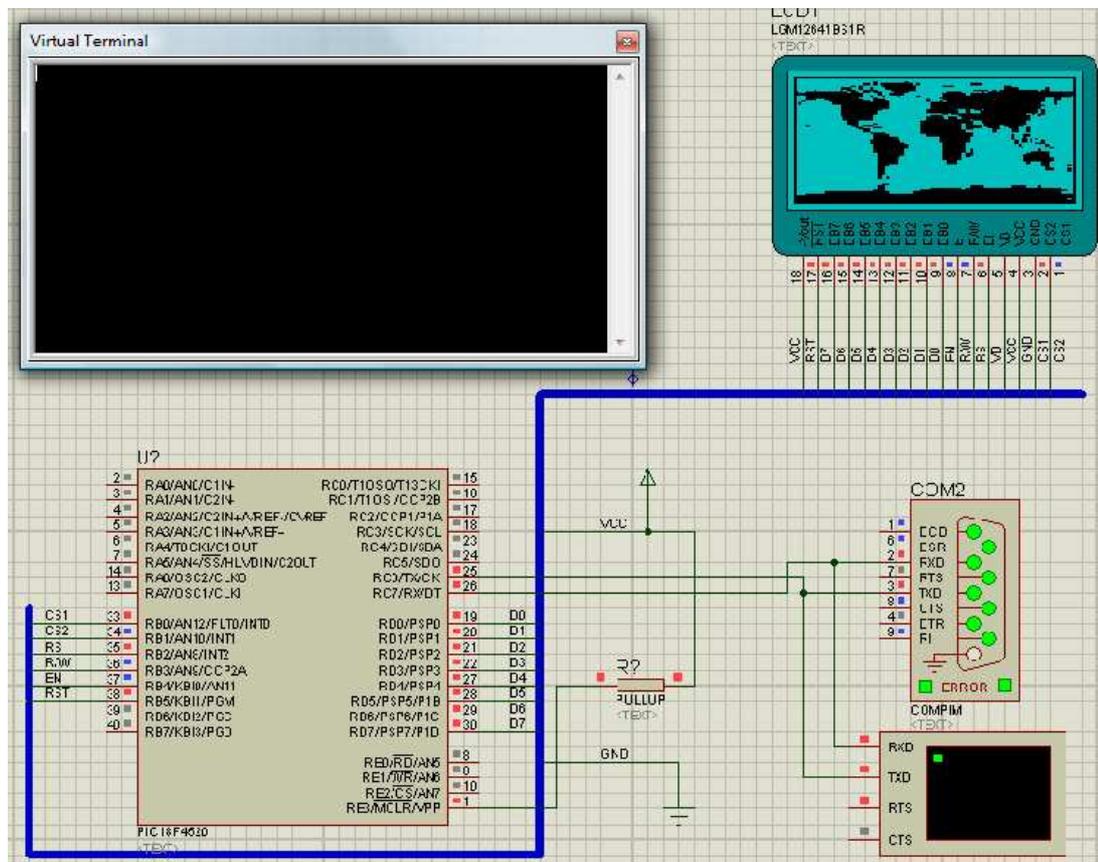


Figura 4.5 Simulación en ejecución con Proteus

4.2.2 Proyecto en Proteus

Para realizar esta simulación se reutilizó la simulación del ejemplo Mapamundi, pero con los siguientes cambios:

Se tomó una fotografía de la vista superior del edificio principal de la FIEC, ver figura 4.6.



Figura 4.6 Croquis de la planta alta del edificio principal FIEC

Se convirtió la imagen anterior a formato Bitmap usando el programa Paint de Accesorios de Windows, luego se la modificó hasta obtener la figura 4.7.



Figura 4.7 Croquis ya convertido a Bitmap

Una vez definida la imagen que serviría como mapa se usó el programa GLCD Bitmap Editor para obtener el mapa de bits que se cargará en la pantalla GLCD.

Se tomaron lecturas de cuatro puntos referenciales del segundo piso del edificio principal de la FIEC, estos puntos fueron los siguientes con sus respectivas lecturas.

Laboratorio de Microcontroladores

\$GPRMC,162148.00,A,0208.69400,S,07958.07360,W,0.636,293.91,15
0410,,A*6B

\$GPVTG,293.91,T,,M,0.636,N,1.179,K,A*30

\$GPGGA,162148.00,0208.69400,S,07958.07360,W,1,06,3.83,81.7,M,1
0.8,M,,*62

\$GPGSA,A,3,29,27,30,12,21,14,,,,,,,,,6.83,3.83,5.66*0A

\$GPGSV,4,1,16,01,12,216,26,02,03,138,,05,01,087,,09,46,052,23*75

\$GPGSV,4,2,16,12,23,136,27,14,33,262,32,15,03,031,,18,19,347,19*75

\$GPGSV,4,3,16,21,46,346,15,22,07,316,,24,40,345,,26,35,355,*7F

\$GPGSV,4,4,16,27,34,045,16,29,47,202,28,30,27,172,37,31,02,211,*7

C

\$GPGLL,0208.69400,S,07958.07360,W,162148.00,A,A*6E

Laboratorio 1 de Programas Utilitarios

\$GPRMC,175447.00,A,0208.69105,S,07958.05004,W,0.704,325.60,15
0410,,A*66

\$GPVTG,325.60,T,,M,0.704,N,1.304,K,A*3A

\$GPGGA,175447.00,0208.69105,S,07958.05004,W,1,09,1.10,94.5,M,1
0.8,M,,*6C

\$GPGSA,A,3,21,24,30,31,22,29,09,12,14,,,,,2.05,1.10,1.73*00

\$GPGSV,4,1,14,01,30,183,37,06,05,273,,09,08,030,14,12,24,089,28*79

\$GPGSV,4,2,14,14,27,313,10,18,35,026,16,21,84,193,25,22,22,349,16*
7C

\$GPGSV,4,3,14,24,84,303,21,26,07,025,,27,01,031,,29,21,164,27*7E

\$GPGSV,4,4,14,30,41,125,35,31,39,221,29*78

\$GPGLL,0208.69105,S,07958.05004,W,175447.00,A,A*61

Laboratorio de Ingeniería de Software

\$GPRMC,175854.00,A,0208.66926,S,07958.05327,W,0.236,283.24,15
0410,,A*65

\$GPVTG,283.24,T,,M,0.236,N,0.437,K,A*35

\$GPGGA,175854.00,0208.66926,S,07958.05327,W,1,09,1.10,69.6,M,1
0.8,M,,*67

\$GPGSA,A,3,24,21,30,22,29,09,31,12,14,,,,,1.80,1.10,1.43*0D

\$GPGSV,4,1,13,01,31,181,,06,05,271,17,09,07,030,27,12,24,087,36*73

\$GPGSV,4,2,13,14,26,315,19,18,36,028,28,21,82,186,22,22,23,350,26*
7A

\$GPGSV,4,3,13,24,85,282,25,26,07,027,19,29,20,163,27,30,42,122,31*
78

\$GPGSV,4,4,13,31,41,223,25*48

\$GPGLL,0208.66926,S,07958.05327,W,175854.00,A,A*6B

Laboratorio de CISCO

\$GPRMC,180213.00,A,0208.69331,S,07958.05457,W,0.414,298.38,15
0410,,A*64

\$GPVTG,298.38,T,,M,0.414,N,0.768,K,A*3D

\$GPGGA,180213.00,0208.69331,S,07958.05457,W,1,09,0.94,80.0,M,1
0.8,M,,*6B

\$GPGSA,A,3,31,22,30,24,21,29,18,12,14,,,,,3.08,0.94,2.93*0C

\$GPGSV,4,1,13,01,32,180,,06,06,270,25,09,05,030,,12,23,085,39*7F

\$GPGSV,4,2,13,14,26,317,32,18,37,029,27,21,80,183,21,22,24,351,21*

7B

\$GPGSV,4,3,13,24,85,260,21,26,06,028,,29,20,161,31,30,42,120,31*71

\$GPGSV,4,4,13,31,42,224,32*4A

\$GPGLL,0208.69331,S,07958.05457,W,180213.00,A,A*6B

Con estas medidas se calcularon los mínimos y máximos de X y Y para la pantalla GLCD.

Con estos cambios se procedió a implementar los cambios en el programa, se compiló esta solución y se cargo esta programación al PIC 18F4520 de la simulación en Proteus obteniendo como resultado la figura 4.8.

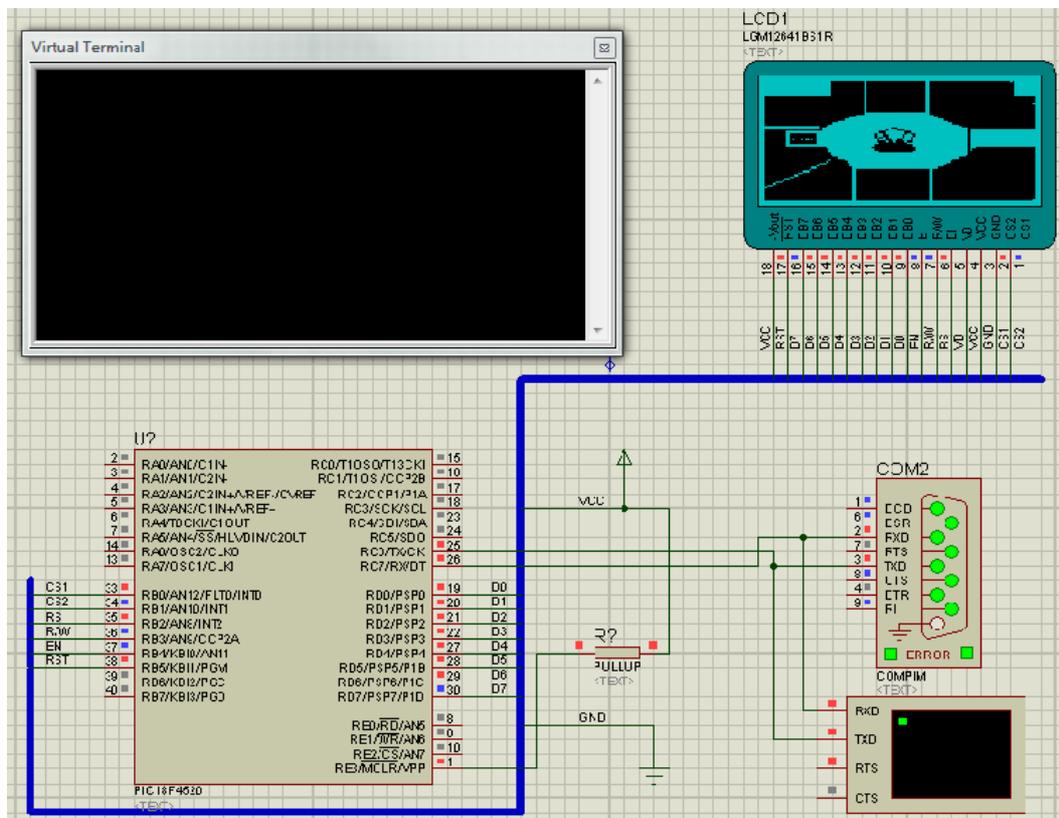


Figura 4.8 Simulación utilizando croquis en Bitmap

La prueba con la tarjeta se la realizó tomando como referencia el laboratorio de microcontroladores, obteniendo como resultado un punto donde se encuentra el laboratorio, las lecturas que presentaba la tarjeta iban variando constantemente ero estas se mantenían dentro del laboratorio y estos se mantenían dentro del rango aceptado que es de aproximadamente de 5 metros a la redonda.

CONCLUSIONES

1. Al realizar este proyecto pudimos darnos cuenta de la capacidad que tenemos para utilizar sistemas de posicionamiento global vinculado con microcontroladores para poder supervisar diversos procesos de la vida diaria.
2. Con la implementación de este proyecto se logró establecer la ubicación de un objeto dentro de un rango aceptable de error, aproximadamente de 5 metros a la redonda. Este error se concluye puede ser mejorado con un equipo que proporcione datos más precisos, es decir, una cadena de datos con más decimales.
3. Ya con nuestra práctica 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.
4. Se pudo concluir además que para fines educativos, esta solución es muy conveniente en el tema económico, comparando otras tarjetas de desarrollo existentes en el mercado, esta tiene más funcionalidades, como se ha descrito en el presente trabajo.

RECOMENDACIONES

1. Es necesario que el usuario del módulo GPS revise primero la documentación de los componentes que forman parte de la tarjeta SmartGPS, ya que esto le ayudará en el uso de la misma y así proteger la integridad de la tarjeta. Una vez que se tenga el conocimiento de los componentes que conforman la tarjeta 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 me permita visualizar la información del GPS.
3. La modo de alimentación de la tarjeta 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. Es recomendable investigar el funcionamiento del programa GoogleMaps, 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 el módulo GPS.

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 del módulo GPS es muy sencilla, además de que provee de herramientas muy valiosas para realizar simulaciones de los datos que envía el módulo GPS.

ANEXOS

ANEXO 1

Hoja de datos técnicos módulo GPS LEA5S



LEA-5 Module Series

u-blox 5 GPS and GALILEO Receivers

Preliminary Data

Overview

The LEA-5 module series brings the high performance of the u-blox 5 positioning engine to the industry standard LEA form factor. These versatile, stand-alone receivers combine an extensive array of features with flexible connectivity options. Their ease of integration results in fast times-to-market for a wide range of automotive, consumer and industrial applications with strict size and cost requirements.



22.4 x 17.0mm

*your position
is our focus*



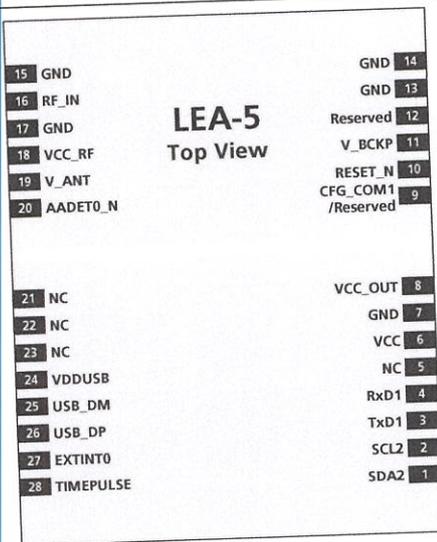
Highlights

- 50-channel u-blox 5 engine with over 1 million effective correlators
- <1 second Time To First Fix for Hot and Aided Starts
- -160dBm SuperSense® acquisition and tracking sensitivity
- Accelerated startup at weak signals with KickStart feature
- Supports AssistNow Online and AssistNow Offline A-GPS services; OMA SUPL compliant
- High immunity to jamming
- Hybrid GPS, GALILEO and SBAS (WAAS, EGNOS, MSAS, GAGAN) engine
- 4 Hz position update rate
- Easy migration from LEA-4 modules

Features

	Power	Size	Memory	Function	Antenna	Input / Output
	Voltage Range (V)	Thickness (mm)	Programmable (Flash) FW Update	KickStart Dead Reckoning Raw Data Precision Timing	Antenna Supply Antenna Supervisor	UART USB SPI DDC (I ² C compliant) Reset Input Configuration Pin
LEA-5H	2.7-3.6	3.0	•	•	• •	1 1 1 •
LEA-5S				•	• •	1 1 1 • 1
LEA-5A					• •	1 1 1 • 1

Mechanical Data



Dimensions 22.4mm x 17.0mm

Weight 2.1 g

Receiver Performance Data

Receiver Type	50-channel u-blox 5 engine GPS L1 C/A code GALILEO L1 Open Service (with upgrade) SBAS: WAAS, EGNOS, MSAS, GAGAN	
Max. Update Rate	4 Hz	
Accuracy¹	Position	2.5 m CEP
	SBAS	2.0 m CEP
Acquisition¹	Cold starts:	29 s
	Warm starts:	29 s
	Aided starts ² :	<1 s
	Hot starts:	<1 s
Sensitivity	Acquisition:	-160 dBm
	Tracking:	-160 dBm
	Cold starts:	-145 dBm
Multipath Suppression	Intelligent multipath detection and suppression	
A-GPS	Supports AssistNow Online and AssistNow Offline, OMA SUPL compliant	
Operational Limits	Velocity:	515 m/s (1000 knots)

¹ All SV @ -130 dBm

² Dependent on aiding data connection speed and latency

Interfaces

Serial Interfaces	1 UART 1 USB V2.0 Full Speed 12 Mbit/s 1 DDC (I ² C compliant) 1 SPI
Digital I/O	Configurable time pulse 1 EXTINT input 1 Reset 1 Configuration Pin (LEA-5A, LEA-5S)
Serial and I/O Voltages	3 V levels
Protocols	NMEA, UBX binary

Electrical Data

Power Supply	2.7 to 3.6 V
Power Consumption	120 mW @ 3.0V
Backup Power	1.3 V to 4.8 V, 30µA
Antenna Type	Active and Passive
Antenna Power	External or Internal VCC_RF
Antenna Supervision	Integrated short-circuit detection and antenna shutdown, open circuit detection is supported with AADET_N input and little external circuitry

Environmental Data

Operating Temp.	-40°C to 85°C
Storage Temp.	-40°C to 85°C

Support Products

EVK-5H	u-blox 5 Evaluation Kit An easy-to-use kit to get familiar with u-blox 5 positioning technology, and to evaluate functionality and visualize GPS performance.
---------------	--

Ordering Information

LEA-5H-0	LEA-5H – u-blox 5 GPS/GALILEO Module
LEA-5S-0	LEA-5S – u-blox 5 GPS Module
LEA-5A-0	LEA-5A – u-blox 5 GPS Module
	Available as samples and tape on reel (250 pieces)

Legal Notice

u-blox reserves all rights to this document and the information contained herein. Products, names, logos and designs described herein may in whole or in part be subject to intellectual property rights. Reproduction, use, modification or disclosure to third parties of this document or any part thereof without the express permission of u-blox is strictly prohibited.

The information contained herein is provided "as is". No warranty of any kind, either express or implied, is made in relation to the accuracy, reliability, fitness for a particular purpose or content of this document. This document may be revised by u-blox at any time. For most recent documents, please visit www.u-blox.com.

Copyright © 2007, u-blox AG

GPS.G5-MS5-07071-P4 12-12-2007

u-blox AG
Zürcherstrasse 68
8800 Thalwil
Switzerland
www.u-blox.com

Phone: +41 44 722 7444
Fax: +41 44 722 7447
info@u-blox.com

ANEXO 2

Hoja de datos técnicos PIC18F452



PIC18FXX2

28/40-pin High Performance, Enhanced FLASH Microcontrollers with 10-Bit A/D

High Performance RISC CPU:

- C compiler optimized architecture/instruction set
 - Source code compatible with the PIC16 and PIC17 instruction sets
- Linear program memory addressing to 32 Kbytes
- Linear data memory addressing to 1.5 Kbytes

Device	On-Chip Program Memory		On-Chip RAM (bytes)	Data EEPROM (bytes)
	FLASH (bytes)	# Single Word Instructions		
PIC18F242	16K	8192	768	256
PIC18F252	32K	16384	1536	256
PIC18F442	16K	8192	768	256
PIC18F452	32K	16384	1536	256

- Up to 10 MIPS operation:
 - DC - 40 MHz osc./clock input
 - 4 MHz - 10 MHz osc./clock input with PLL active
- 16-bit wide instructions, 8-bit wide data path
- Priority levels for interrupts
- 8 x 8 Single Cycle Hardware Multiplier

Peripheral Features:

- High current sink/source 25 mA/25 mA
- Three external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter with 8-bit programmable prescaler
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter with 8-bit period register (time-base for PWM)
- Timer3 module: 16-bit timer/counter
- Secondary oscillator clock option - Timer1/Timer3
- Two Capture/Compare/PWM (CCP) modules. CCP pins that can be configured as:
 - Capture input: capture is 16-bit, max. resolution 6.25 ns ($T_{CY}/16$)
 - Compare is 16-bit, max. resolution 100 ns (T_{CY})
 - PWM output: PWM resolution is 1- to 10-bit, max. PWM freq. @: 8-bit resolution = 156 kHz
10-bit resolution = 39 kHz
- Master Synchronous Serial Port (MSSP) module, Two modes of operation:
 - 3-wire SPI™ (supports all 4 SPI modes)
 - I²C™ Master and Slave mode

Peripheral Features (Continued):

- Addressable USART module:
 - Supports RS-485 and RS-232
- Parallel Slave Port (PSP) module

Analog Features:

- Compatible 10-bit Analog-to-Digital Converter module (A/D) with:
 - Fast sampling rate
 - Conversion available during SLEEP
 - Linearity ≤ 1 LSB
- Programmable Low Voltage Detection (PLVD)
 - Supports interrupt on-Low Voltage Detection
- Programmable Brown-out Reset (BOR)

Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced FLASH program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory
- FLASH/Data EEPROM Retention: > 40 years
- Self-reprogrammable under software control
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own On-Chip RC Oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options including:
 - 4X Phase Lock Loop (of primary oscillator)
 - Secondary Oscillator (32 kHz) clock input
- Single supply 5V In-Circuit Serial Programming™ (ICSP™) via two pins
- In-Circuit Debug (ICD) via two pins

CMOS Technology:

- Low power, high speed FLASH/EEPROM technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Industrial and Extended temperature ranges
- Low power consumption:
 - < 1.6 mA typical @ 5V, 4 MHz
 - 25 μ A typical @ 3V, 32 kHz
 - < 0.2 μ A typical standby current

ANEXO 3

Hoja de datos técnicos GLCD 128X64

82.8±0.5

P2.54*(25-1)=60.96

25-φ1.0

10.92

1

25

70±0.5

58.2±0.2

44(V.A)

38.36

128 * 64 dots

65±0.2

56.28

62(V.A.)

72.8±0.2

73.5±0.2

4-φ2.5

14.8(MAX)

10.2

11.4±0.3

1.6

0.40

0.56

0.04

0.04

TOLERANCES UNLESS OTHERWISE STATED

XX.X ± 0.20 X.X ± 0.10

PERFORMANCE FEATURES

LCD TYPE	STN, Transflective
Viewing Angle	6:00
Backlight	LED SideLight(R, G, B)
Operation temperature	-20° C ~ +70° C
Storage temperature	-25° C ~ +80° C
Controller	S6B0107 S6B0108
Operation Voltage of LCM	+5.0V

PIN	SIGNAL	PIN	SIGNAL	PIN	SIGNAL
1	/CS1	10	DB1	19	LED R+
2	/CS2	11	DB2	20	LED G+
3	VSS	12	DB3	21	LED B+
4	VDD	13	DB4	22	Y-
5	V0	14	DB5	23	X-
6	D/I	15	DB6	24	Y+
7	R/W	16	DB7	25	X+
8	E	17	/RST		
9	DB0	18	VEE		

UNIT: mm

REVISION RECORD

VER	Date	Modify Contents	Name	Confirm
2				
3				
4				
5				
6				

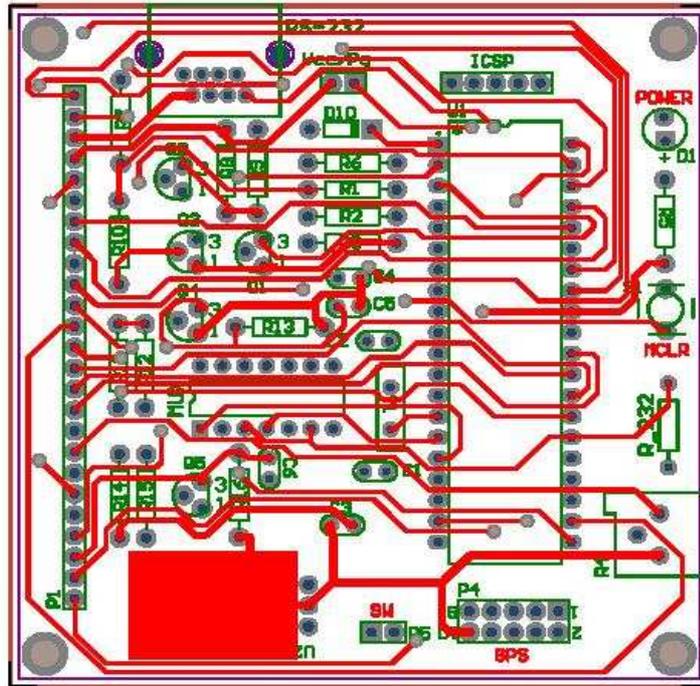
Drawn: J.J.LIN
 Checked: C.cheung
 Approved:
 Model Name: DM1288-4
 Date: 2007-11-08



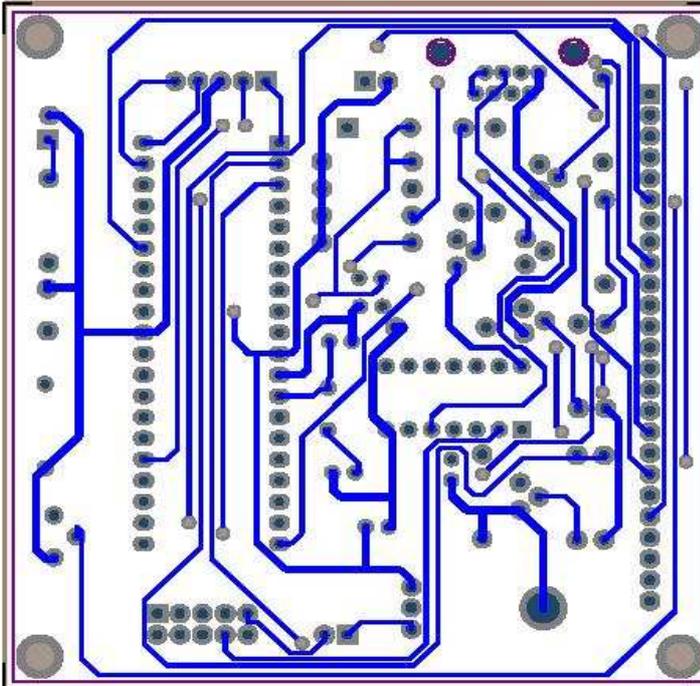
ANEXO 4

Esquemático y PCB del circuito

PCB Cara A



PCB Cara B



ANEXO 5

Lista de Materiales y Presupuesto

Cant.	Descripción	Precio U	Precio T
1	Tarjeta SmartGPS	\$ 97,00	\$ 97,00
1	Pantalla GLCD 128X64 RGB + Panel táctil	\$ 55,00	\$ 55,00
1	PIC18F452 microcontrolador	\$ 9,00	\$ 9,00
15	Resistores película de carbón 1/2W	\$ 0,03	\$ 0,45
5	Transistores propósito general 2N3904/2N3906	\$ 0,06	\$ 0,30
1	Oscilador de cuarzo de 10Mhz	\$ 0,70	\$ 0,70
1	Diodo 1N4148	\$ 0,05	\$ 0,05
5	Capacitores cerámicos 50V	\$ 0,06	\$ 0,30
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	\$ 0,10	\$ 0,10
1	Switch	\$ 0,20	\$ 0,20
1	CI Buffer 3 estados 74125	\$ 3,00	\$ 3,00
1	Fabricación PCB en placa de fibra con antisolder y serigrafía	\$ 50,00	\$50,00
		TOTAL	\$ 218,05

BIBLIOGRAFÍA

- [1]. U-blox; Data Sheet LEA-5X;
<http://www.u-blox.com/en/download-center.html> ; **Fecha de consulta:**
20/Abril/2010.
- [2]. Microchip, Data Sheet PIC18F4520;
<http://ww1.microchip.com/downloads/en/DeviceDoc/39631a.pdf> **Fecha de consulta:** 20/Abril/2010.
- [3]. Mikroelektronika, Diagrama esquemático de la tarjeta Smart GPS v1.01;
http://www.mikroe.com/pdf/smartgps_schematic_v101.pdf ; Fecha de consulta: 22/Abril/2010.
- [4]. Mikroelektronika; Presentación Tarjeta Smart GPS;
http://www.mikroe.com/eng/downloads/get/37/mikrobasic_pic_pro_manual_v101.pdf ; **Fecha de consulta:** 22/Abril/2010.
- [5]. 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:** 23/Abril/2010.
- [6]. Mikroelektronika; Guía de Referencia MikroBasic;
http://www.mikroe.com/pdf/mikrobasic/basic_syntax_v101.pdf ; **Fecha de consulta:** 23/Abril/2010.
- [7]. GPS – NMEA sentence information;
<http://aprs.gids.nl/nmea/> ; **Fecha de consulta:** 23/Abril/2010.
- [8]. Mikroelektronika; Presentación Pantalla Táctil;
http://www.mikroe.com/eng/downloads/get/468/es_mikroe_article_basic_avr_01_09.pdf ; **Fecha de consulta:** 25/Abril/2010.
- [9]. 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:** 30/Abril/2010.