

## Receptor GPS 28500

### 1.- INTRODUCCION

Se trata del módulo receptor de GPS de altas prestaciones y bajo coste modelo PMB-648 de la firma POLSTAR (muy similar a los modelos PMB-688 y PMB-248) y basado en el chipset de SiRFstar III. Con su antena integrada es capaz de realizar un seguimiento de hasta 20 satélites proporcionando datos según la norma NMEA 0183 para aplicaciones de navegación, robótica, telemetría o experimentación. Incluye batería interna recargable para el mantenimiento de la fecha y hora así como un cable para las conexiones de alimentación y comunicaciones serie tanto en TTL como en RS-232. Ver la figura 1.

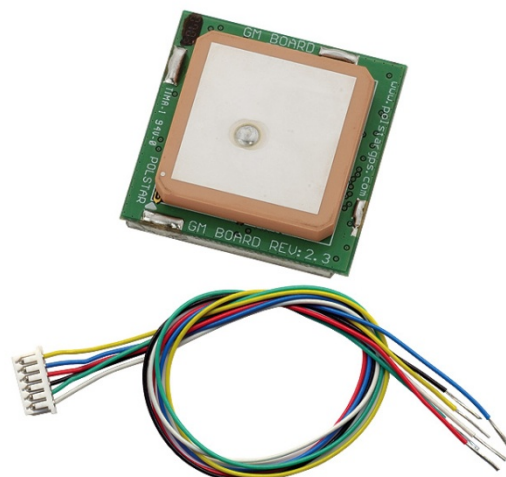


Figura 1. El módulo receptor GPS 28500

### 2.- CARACTERISTICAS

Las características más importantes de este receptor GPS se resumen en la siguiente tabla:

PARAMETRO	DESCRIPCION
Chip Set	SiRFstar III
Recepción	Seguimiento de hasta 20 satélites
Precisión	Posición: Aproximadamente 5 metros Velocidad: 0.1 m/s Tiempo: +/- 1 $\mu$ S
Tiempo de adquisición	Arranque en frío: 42 seg. Aprox. Arranque en caliente: entre 1 seg. y 38 seg. Aprox.
Sensibilidad	Adquisición: -148 dBm Seguimiento: -159 dBm
Dinámica	Altitud: 18000 m (máx.) Velocidad: 515 m/s (máx.) Aceleración: +/- 4g (máx.)
Actualización	1 actualización por segundo
Puerto serie	TTL y RS-232
Baudios	4800
Mensajes de salida	Tramas GGA, GSV, GSA y RMC según NMEA 0183
Alimentación	Desde 3.3 hasta 5 Vcc
Consumo	Típico 65mA a 5Vcc
LED	Alimentación ON/OFF y navegación
Rango de temperatura	-20°C hasta + 70°C
Dimensiones	32 x 32 mm

### 3.- APLICACIONES

Entre las más relevantes caben citar las siguientes:

- Navegación de vehículos terrestres
- Navegación de vehículos marinos
- Navegación personal
- Control de flotas de vehículos
- Servicios de localización
- Sistemas de seguimiento
- Aplicaciones de gestión de rutas y mapas

### 4.- PATILLAJE

Se muestra en la figura 2 y sus conexiones se resumen en la tabla adjunta

PIN	NOMBRE	DESCRIPCION
1	TTL TX	Trasmisión de datos con niveles TTL
2	TTL RX	Recepción de datos con niveles TTL
3	Vcc	Positivo de alimentación
4	GND	Tierra de alimentación
5	RS-232 TX	Transmisión de datos en RS-232
6	RS-232 RX	Recepción de datos en RS-232

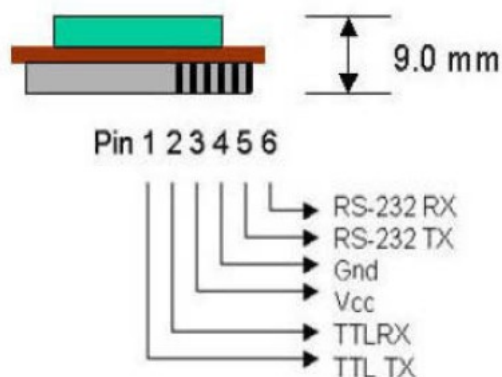


Figura 2. Distribución de señales

### 5. PROTOCOLO NMEA 0183

Se trata de un protocolo definido y controlado por la organización estadounidense National Marine Electronics Association y es un protocolo a través del cual la mayoría de los receptores GPS y otros instrumentos de navegación y/o marítimos se pueden comunicar entre sí y entre otros sistemas como controladores, PC's, etc..

Existe abundante documentación sobre este protocolo, parte de la cual se adjunta en el CDROM. Lo más destacable es que la información que se transfiere en este protocolo desde, por ejemplo un GPS hasta un PC, consta de varias tramas. En el caso concreto del receptor GPS 28500 objeto del presente documento, cada vez que se actualiza dicho receptor (a cada segundo), transmite 4 tramas del protocolo NMEA 0183:

- Trama GGA: Sistema global de posicionamiento
- Trama GSV: Número de mensajes GSV que le siguen, elevación azimuth, etc..
- Trama GSA: Modos de operación del receptor GPS
- Trama RMC: Datos mínimo recomendados

De ellas, las que nos interesan y vamos a emplear en los diferentes ejemplos son la trama GGA y la trama RMC.

#### 5.1 Trama GGA

Comienza siempre mediante los caracteres '\$GPGGA,' y consta a su vez de 15 campos que se separan entre sí mediante ','. Se detallan a continuación:

\$GPGGA,1,2,3,4,5,6,7,8,9,A,B,C,D,E\*F

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
  - A.- Unidades de altitud: M (metros)
  - B.- Separación geoidal: xxx.x
  - C.- Unidades de separación: M (metros)
  - D.- Tiempo en segundos desde la última actualización:xx
  - E.- ID de referencia de la estación
  - F.- Checksum: \*xx

De todos los campos de la trama GGA, en nuestros ejemplos, vamos a emplear los siguientes campos:  
1, 2, 3, 4, 5, 6, 7, 9 y A

## 5.2 Trama RMC

Comienza siempre con los caracteres '\$GPRMC,' y consta a su vez de 12 campos que se separan entre sí mediante ','. Se detallan a continuación:

\$GPRMC, 1,2,3,4,5,6,7,8,9,A,B,\*C

- 1.- Hora UTC (Tiempo Universal Coordinado) en formato: hhmmss.ss
- 2.- Estado de los datos: A=Activo; V=nulo
- 3.- Latitud en formato: ggmm.ssss
- 4.- Orientación en latitud: N (norte) o S (sur)
- 5.- Longitud en formato: gggmm.ssss
- 6.- Orientación en longitud: E (este) o W (oeste)
- 7.- Velocidad en nudos: xxx.x
- 8.- Calidad de seguimiento en grados:xxx.x
- 9.- Fecha UTC en formato: ddmmaa
- A.- Variación magnética en grados
- B.- Orientación de la variación: E (este) o W (oeste)
- C.- Checksum: \*xx

De todos los campos de la trama RMC, en nuestro ejemplos, tan sólo vamos a emplear los campos 7 y 9

En el protocolo NMEA 0183 todos los bytes transferidos son caracteres ASCII. La comunicación se realiza a 4800 baudios, 8 bits de datos, sin paridad y 1 bit de stop. Todas las tramas comienza con los caracteres \$GP y finalizan mediante un retorno de carro (0x0d) y avance de línea (0x0A).

## 6. CONEXIONES

En este apartado se exponen una serie de consideraciones relacionadas con la conexión del módulo receptor GPS.

## 6.1 Preparando las conexiones

En primer lugar recomendamos soldar los cables que acompañan al módulo GPS a un conector de 4 pines con paso normalizado de 2.54 mm, tal y como se muestra en la figura 3. Esto nos permitirá poder conectar el GPS en cualquier tarjeta de prototipos o boards de una forma rápida y cómoda. Únicamente empleamos 4 señales: TTL TX, TTL RX, VCC y GND.

Aconsejamos hacer las soldaduras con el cable desconectado respecto al módulo para evitar calentamientos innecesarios en su circuitería interna

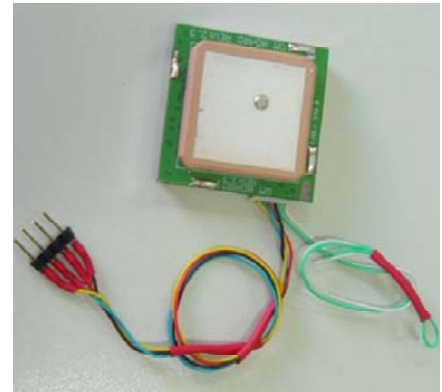
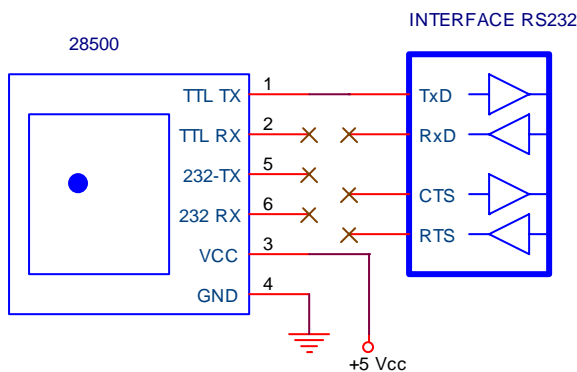


Figura 3. Soldando un conector de 4 pines

## 6.2 Primera toma de contacto



Una primera toma de contacto con el módulo receptor GPS puede consistir en conectarlo directamente con el canal serie de un PC. Empleando el laboratorio USB-PIC'School el interface se puede hacer sencillamente tal y como se muestra en el esquema de la figura 4. Hay que alimentar el módulo GPS con los +5VCC y GND del propio

laboratorio y conectar la señal TTL TX con la línea TxD del interface RS232. Ahora basta conectar el laboratorio USB-PIC'School con un canal serie libre del PC.

Figura 4. Interface del GPS con el canal serie de laboratorio USB-PIC'School

En caso de no disponer del laboratorio, podemos montar el clásico circuito de adaptación de niveles TTL a niveles RS-232. Para ello empleamos el típico circuito MAX232 como el mostrado en la figura 5.

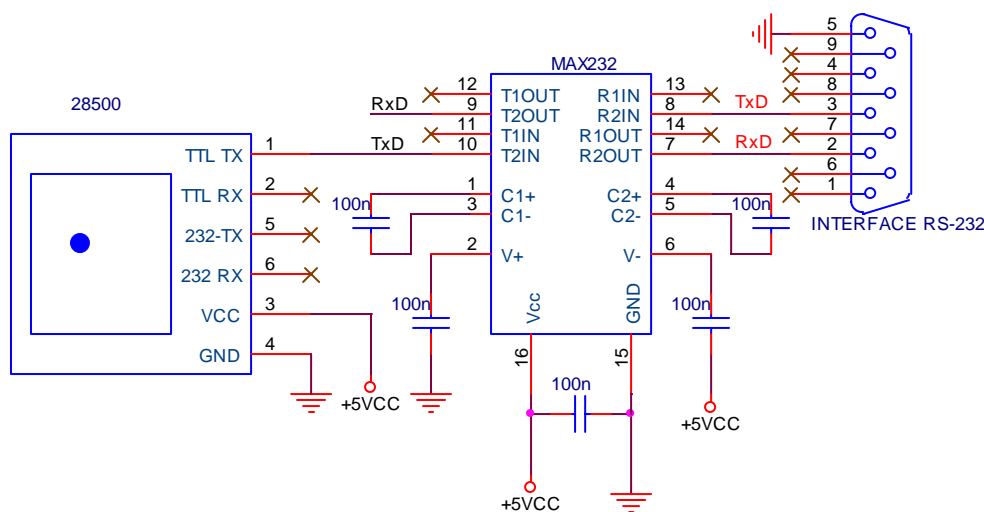
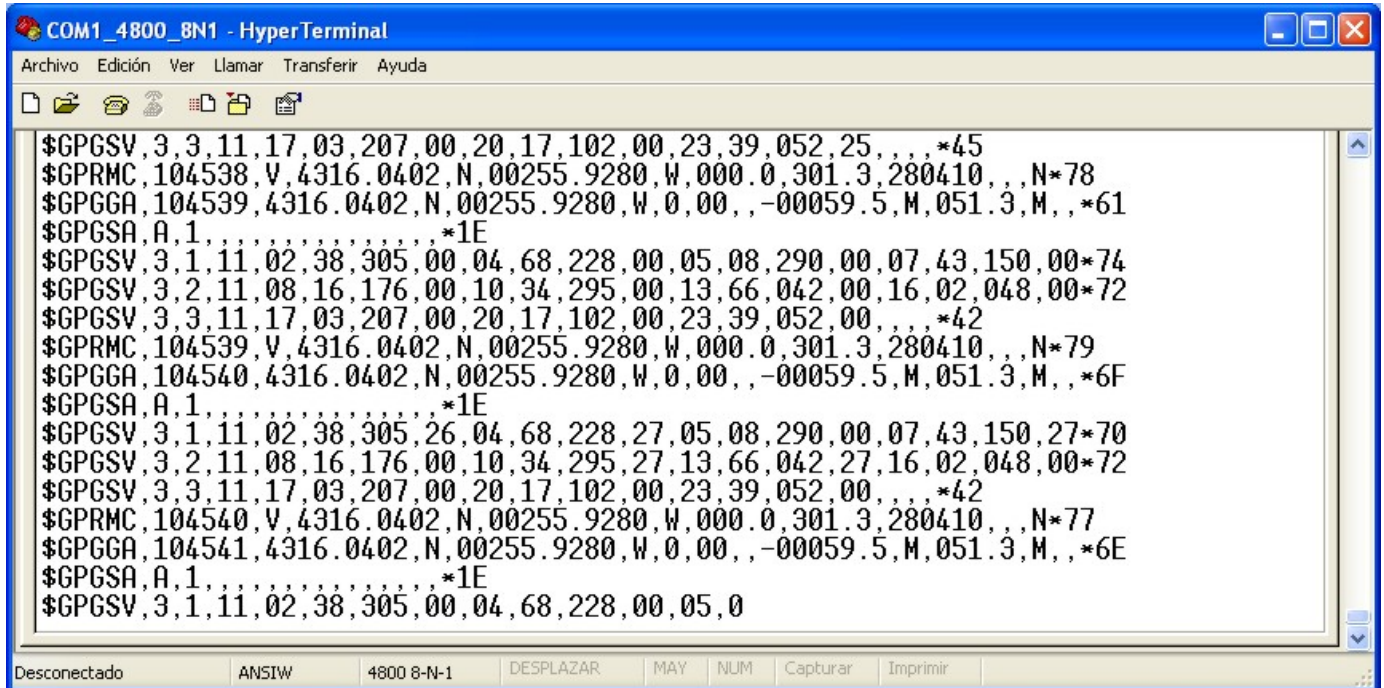


Figura 5. Circuito de interface RS-232

Mediante un simple software de comunicaciones, como puede ser el HyperTerminal de Windows, podemos ver toda la información que nos transmite el GPS de forma similar a la mostrada en la figura 6



```
COM1_4800_BN1 - HyperTerminal
Archivo Edición Ver Llamar Transferir Ayuda
$GPGSV,3,3,11,17,03,207,00,20,17,102,00,23,39,052,25,,,,*45
$GPRMC,104538,V,4316.0402,N,00255.9280,W,000.0,301.3,280410,,,N*78
$GPGGA,104539,4316.0402,N,00255.9280,W,0,00,,-00059.5,M,051.3,M,,*61
$GPGSA,A,1,,,,,,,,,,,,,*1E
$GPGSV,3,1,11,02,38,305,00,04,68,228,00,05,08,290,00,07,43,150,00*74
$GPGSV,3,2,11,08,16,176,00,10,34,295,00,13,66,042,00,16,02,048,00*72
$GPGSV,3,3,11,17,03,207,00,20,17,102,00,23,39,052,00,,,,*42
$GPRMC,104539,V,4316.0402,N,00255.9280,W,000.0,301.3,280410,,,N*79
$GPGGA,104540,4316.0402,N,00255.9280,W,0,00,,-00059.5,M,051.3,M,,*6F
$GPGSA,A,1,,,,,,,,,,,,,*1E
$GPGSV,3,1,11,02,38,305,26,04,68,228,27,05,08,290,00,07,43,150,27*70
$GPGSV,3,2,11,08,16,176,00,10,34,295,27,13,66,042,27,16,02,048,00*72
$GPGSV,3,3,11,17,03,207,00,20,17,102,00,23,39,052,00,,,,*42
$GPRMC,104540,V,4316.0402,N,00255.9280,W,000.0,301.3,280410,,,N*77
$GPGGA,104541,4316.0402,N,00255.9280,W,0,00,,-00059.5,M,051.3,M,,*6E
$GPGSA,A,1,,,,,,,,,,,,,*1E
$GPGSV,3,1,11,02,38,305,00,04,68,228,00,05,0
```

Figura 6. Visualización de las tramas enviadas por el módulo GPS 28500

La transferencia de datos comienza con la trama \$GPGGA. A continuación le sigue una trama \$GPGSA seguida de una o más tramas \$GPGSV. Finaliza con la trama \$GPRMC. Toda esta transferencia se repite a intervalos de 1 segundo.

La tarea de nuestro controlador consistirá en decodificar y recoger las tramas que nos interese así como los campos que precisemos dentro de cada trama. En los ejemplos propuesto se decodificarán únicamente las tramas \$GPGGA y \$GPRMC.

### 6.3 Problemas durante la experimentación

El módulo receptor 28500, al igual que cualquier otro receptor GPS, sólo son capaces de sintonizar con los diferentes satélites cuando se encuentra en un área exterior despejada. Es decir, un receptor GPS en un espacio interior como una habitación o un laboratorio, es muy difícil que pueda captar la señal de ningún satélite.

Esto supone un problema ya que normalmente nuestro laboratorio o taller de experimentación se encuentra normalmente en el interior de un edificio. En Ingeniería de Microsistemas Programados conscientes y víctimas del problema, sugerimos prolongar las conexiones entre el módulo GPS y el laboratorio USB-PIC'School. De esta forma podremos sacar al exterior, por una ventana, nuestro receptor, mientras que el laboratorio de pruebas se encuentra en el interior. Para ello, tal y como se muestra en la figura 7 hemos empleado una manguera de 4 hilos y unos 6 m de longitud que transporta las señales de +VCC, GND y TTL TX.

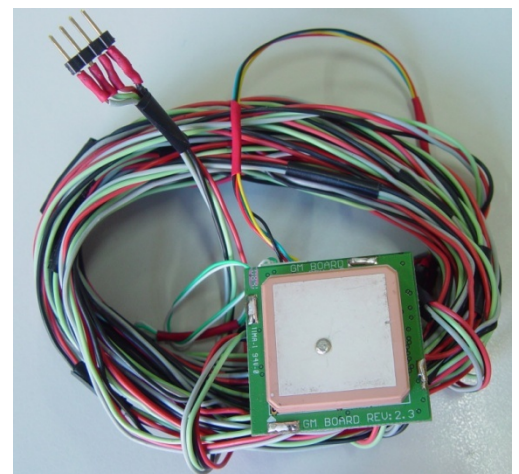


Figura 7. Prolongación de las conexiones entre el receptor GPS y el laboratorio de prácticas

Esto a su vez nos crea un nuevo problema. Cuanto más largas sean las conexiones entre el receptor GPS y el laboratorio, mayor es la pérdida de calidad de la señal TTL TX que transporta los datos a nuestro controlador. La solución utilizada por Ingeniería de Microsistemas Programados consiste en emplear un sencillo sistema amplificador como el que se muestra en el esquema de la figura 8.

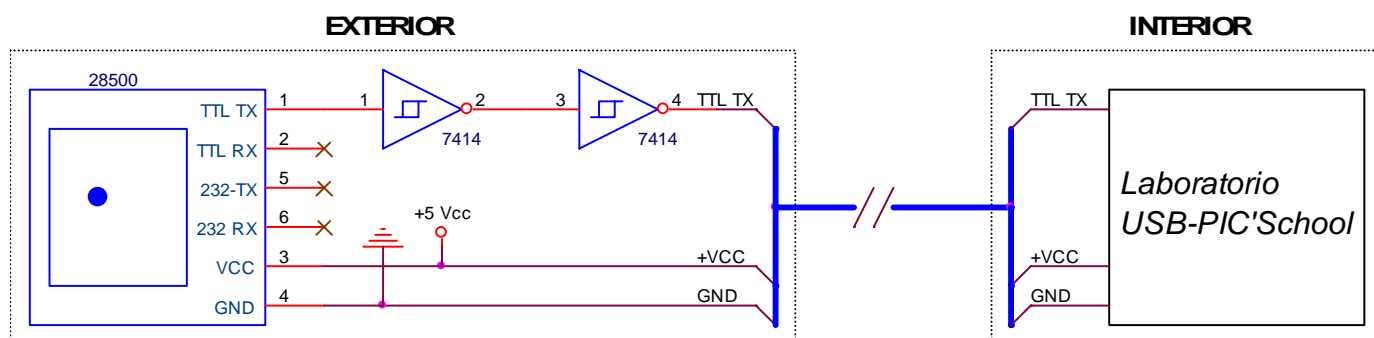


Figura 8. Amplificación de la señal TTL TX

## 7.- LIBRERIAS

En el CDRom, junto con los programas fuente de los ejemplos propiamente dichos, se proporcionan una serie de librerías escritas tanto en ensamblador (\*.INC) como en lenguaje C de alto nivel (\*.H). A su vez cada librería incluye una serie de rutinas o funciones que facilitan la realización de diferentes tareas. Estas librerías se incluyen en los programas fuente mediante la directiva `<include>`.

### 7.1 Librería: `lcd4bitsPIC16`

Contiene rutinas para el control de una pantalla LCD mediante interface de 4 bits y que se usará para visualizar los resultados de los distintos ejemplos.

NOMBRE	PARAM. DE ENTRADA	PARAM. DE SALIDA	DESCRIPCION
<code>UP_LCD</code>	Ninguno	Ninguno	Configura las líneas de E/S para adaptarlas a la pantalla LCD
<code>LCD_INI</code>	Ninguno	Ninguno	Rutina para la inicialización de la pantalla LCD según especificaciones del fabricante
<code>LCD_DATO</code>	W=Contiene el carácter a visualizar	Ninguno	Envía a la pantalla el dato a visualizar en la posición actual del cursor
<code>LCD_REG</code>	W=Contiene el comando a ejecutar por parte de la pantalla LCD	Ninguno	Envía a la pantalla el comando que debe ejecutar

### 7.2 Librería: `RS232LOWPIC16`

El interface entre el host y el módulo receptor GPS 28500 se realiza mediante una única línea y según el protocolo serie estándar. Esta librería contiene funciones que implementan, por software, la transmisión y recepción de datos hacia/desde el módulo. Los ejemplos escritos en C no necesitan de esta librería ya que las funciones de transferencia serie de datos están integradas en el propio compilador.

NOMBRE	PARAM. DE ENTRADA	PARAM. DE SALIDA	DESCRIPCION
<code>RxS</code>	Ninguno	Rxdreg	Espera a recibir un carácter vía serie que quedará almacenado en Rxdreg
<code>TxS</code>	Txdreg	Ninguno	Transmite vía serie el carácter previamente almacenado en Txdreg

### 7.3 Librería: MSE\_Mat\_PIC16.inc

Contiene una serie de funciones que resuelven las operaciones matemáticas más elementales y que son empleadas por algunos de los ejemplos escritos en ensamblador. Los ejemplos escritos en C no necesitan de esta librería ya que la mayor parte de las funciones matemáticas están integradas en el propio compilador.

NOMBRE	PARAM. DE ENTRADA	PARAM. DE SALIDA	DESCRIPCION
<b>Sum_BCD</b>	Mat_Dato_AL+Mat_Dato_BL	Mat_Dato_BL Mat_Dato_AH	Suma dos números BCD (00-99): Mat_Dato_AL + Mat_Dato_BL. El resultado se almacena en Mat_Dato_BL. En Mat_Dato_AH queda, si lo hubiera, el overflow (resultado mayor de 99)
<b>Sub_BCD</b>	Mat_Dato_AL- Mat_Dato_BL		Resta dos números BCD (00-99): Mat_Dato_AL - Mat_Dato_BL. El resultado se almacena en Mat_Dato_AL. En Mat_Dato_BH queda, si lo hubiera, el overflow (A<B).
<b>Sum16</b>	Mat_Dato_BH:Mat_Dato_BL+ Mat_Dato_AH:Mat_Dato_AL	Mat_Dato_BH Mat_Dato_BL C	Suma dos números de 16 bits. Mat_Dato_BH:Mat_Dato_BL+ Mat_Dato_AH:Mat_Dato_AL El resultado se deposita en Mat_Dato_BH:Mat_Dato_BL. Si hay llevada en el bit 16º, C=1
<b>Sub16</b>	Mat_Dato_AH:Mat_Dato_AL- Mat_Dato_BH:Mat_Dato_BL.	Mat_Dato_BH Mat_Dato_BL C	Resta dos números de 16 bits. Mat_Dato_AH:Mat_Dato_AL- Mat_Dato_BH:Mat_Dato_BL. El resultado se almacena en Mat_Dato_BH:Mat_Dato_BL. Si hay llevada en el bit 16º, C=0
<b>Comp16</b>	Mat_Dato_AH:Mat_Dato_AL(A) Mat_Dato_BH:Mat_Dato_BL(B)	C, Z	Compara dos nº de 16 bits contenidos en Mat_Dato_AH:Mat_Dato_AL(A) con Mat_Dato_BH:Mat_Dato_BL(B) Si A>B --> STATUS<C>=1 y <Z>=0; Si A<B --> STATUS<C>=0 y <Z>=0; Si A=B --> STATUS<C>=1 y <Z>=1
<b>Mul8x8</b>	Mat_Dato_AL * Mat_Dato_BL.	Mat_RES2:Mat_RES3	Multiplca dos nºs de 8 bits contenidos en Mat_Dato_AL y Mat_Dato_BL. El resultado de 16 bits se almacena en Mat_RES2 y Mat_RES3 (LSB).
<b>Mul16x16</b>	Mat_Dato_BH:Mat_Dato_BL* Mat_Dato_AH:Mat_Dato_AL	Mat_RES0:Mat_RES3	Multiplca dos números de 16 bits: Mat_Dato_BH:Mat_Dato_BL* Mat_Dato_AH:Mat_Dato_AL El resultado de 32 bits se almacena en Mat_RES0:Mat_RES3 (LSB)
<b>Div16x16</b>	Mat_Dato_AH:Mat_Dato_AL/ Mat_Dato_BH:Mat_Dato_BL	Mat_Dato_AH:Mat_Dato_AL Mat_RES3:Mat_RES2	Esta rutina divide dos números de 16 bits. El dividendo se almacena en Mat_Dato_AH:Mat_Dato_AL y el divisor en Mat_Dato_BH:Mat_Dato_BL. El cociente se almacena en Mat_Dato_AH:Mat_Dato_AL y el resto en Mat_RES3:Mat_RES2.
<b>Bits8_BCD</b>	W	Mat_RES0	Convierte un número binario de 8 bits en el registro W, en 2 dígitos BCD (de 00 a 99). El resultado se almacena en Mat_RES0. P.e. W=0x2A, Mat_RES0=42 (0x2A = 42).
<b>Bits16_BCD</b>	Mat_Dato_AH:Mat_Dato_AL	Mat_RES0 Mat_RES1 Mat_RES2 (LSB)	Esta rutina convierte un número binario de 16 bits situado en Mat_Dato_AH y Mat_Dato_AL y, lo convierte en 5 dígitos BCD que se depositan en las variables Mat_RES0, Mat_RES1 y Mat_RES2, siendo esta última la de menos peso.
<b>BCD_Bits16</b>	Mat_RES0:Mat_RES2 (LSB)	Mat_Dato_AH:Mat_Dato_AL	Convierte un número de 5 dígitos en BCD, en un número de 16 bits. En Mat_RES0:Mat_RES2 (LSB) se encuentra en nº de 5 dígitos. El resultado se almacena en Mat_Dato_AH:Mat_Dato_AL.
<b>BCD_ASCII</b>	Mat_RES0:Mat_RES2 (LSB) Mat_Point	Mat_RES0(MSB):Mat_RES6(LSB)	Convierte un número BCD de hasta 6 dígitos almacenados en Mat_RES0:Mat_RES2 (LSB) En una cadena ASCII que se almacena a partir de Mat_RES0(MSB) hasta Mat_RES6(LSB). La variable Mat_Point indica el lugar donde colocar un punto decimal (0 =ninguno). P.e. Mat_Point=1, el resultado sería .xxxxxx; Mat_Point=3, el resultado sería xx.xxxx

Librería: GPS28500\_PIC16

Contiene funciones específicas para el control del módulo receptor GPS 28500 de Parallax.

NOMBRE	PARAM. DE ENTRADA	PARAM. DE SALIDA	DESCRIPCION
<i>RxGGA</i>		Buffer 0x190	Espera a recibir del GPS la trama GGA que la almacena en un buffer con inicio 0x190 del banco 3. Se obtiene información de los siguientes campos: Hora, Latitud, Longitud, Calidad, Nº de satélites y altura sobre el nivel de mar.
<i>RxRMC</i>		Buffer 0x1C5	Espera a recibir del GPS la trama RMC que se almacena en el buffer con inicio en 0x1C5 del banco 3. Se obtiene información de los siguientes campos: Velocidad y fecha.
<i>RxGPS</i>		Buffer 0x190 Buffer 0x1C5	Esta función recibe del GPS y registra las tramas GGA y RCM con la información de: Hora UTC, latitud, longitud, calidad de la señal, nº de satélites, altura, velocidad y fecha
<i>GPS_Time</i>		GPS_Buffer= hhmss	Devuelve en GPS_Buffer la hora UTC actual en formato hhmss. Si el campo de hora estuviera vacío, el GPS_Buffer se llena con el carácter '-'
<i>GPS_Date</i>		GPS_Buffer= ddmmaa	Devuelve en GPS_Buffer la fecha UTC actual en formato ddmmaa. Si el campo de fecha estuviera vacío, el GPS_Buffer se llena con el carácter '-'
<i>GPS_Signal</i>		GPS_Buffer= '0' o '1'	Devuelve en GPS_Buffer un carácter que expresa el estado de la señal: '1' (0x31)= señal válida; '0' (0x30)= señal no válida
<i>GPS_Sat</i>		GPS_Buffer= xx	Devuelve en GPS_Buffer dos caracteres (xx) que representan el número de satélites visibles en un momento dado
<i>GPS_Alt</i>		GPS_Buffer= xxxxx.x	Devuelve en GPS_Buffer 7 caracteres en formato xxxxx.x que representan la altura en metros respecto al nivel del mar
<i>GPS_Vel_N</i>		GPS_Buffer= xxx.x	Devuelve en GPS_Buffer 5 caracteres en formato xxx.x que representan la velocidad en nudos
<i>GPS_Vel_K</i>		GPS_Buffer= xxx.x	Devuelve en GPS_Buffer 5 caracteres en formato xxx.x que representan la velocidad en km/h. Para ello se multiplican los nudos por la constante 1.85
<i>GPS_Lat_O</i>		GPS_Buffer= N o S	Devuelve en GPS_Buffer 1 carácter ('N' o 'S') que expresa la orientación Norte/Sur en la latitud
<i>GPS_Lat_D</i>		GPS_Buffer=xxxx.xxxx	Devuelve en GPS_Buffer 9 caracteres en formato xxxx.xxxx que representan la latitud en decimal
<i>GPS_Lon_O</i>		GPS_Buffer= O o W	Devuelve en GPS_Buffer 1 carácter ('O' o 'W') que expresa la orientación Este/Oeste en la longitud
<i>GPS_Lon_D</i>		GPS_Buffer=xxxxx.xxxx	Devuelve en GPS_Buffer 10 caracteres en formato xxxxx.xxxx que representan la longitud en decimal
<i>GPS_Lat_S</i>		GPS_Buffer= ggmss.s	Devuelve en GPS_Buffer caracteres en formato ggmss.s que representan la latitud en sexagesimal (gg <sup>o</sup> mm'ss.s")
<i>GPS_Lon_S</i>		GPS_Buffer= gggmss.s	Devuelve en GPS_Buffer caracteres en formato gggmss.s que representan la longitud en sexagesimal (ggg <sup>o</sup> mm'ss.s")

## 8.- EJEMPLOS

Junto con el receptor GPS 28500 se proporciona un CDROM con una serie de ejemplos didácticos desarrollados por Ingeniería de Microsistemas Programados al objeto de facilitar el uso del módulo. Se facilitan las librerías y programas fuente escritos tanto en ensamblador como en lenguaje C de alto nivel para PIC16F886.

Todos los ejemplos tienen algo en común. En todos ellos se emplea la función o rutina RxGPS que se encarga de recibir y decodificar las tramas GGA y RMC. De la trama GGA se almacenan todos los campos sobre un buffer en RAM que comienza en la posición 0x190. De la trama RMC sólo se recogen los campos de velocidad y fecha que se almacenan en otro buffer de RAM cuyo inicio es la posición 0x1C5.

Los distintos ejemplos van haciendo uso del resto de funciones contenidos en la librería GPS28500\_PIC16 para decodificar en particular los distintos campos de esas tramas.

## 8.1 Ejemplo 1: Obtención de fecha y hora

### Objetivos

Leídas las tramas GGA y RMC se extraen y decodifican los correspondientes campos para visualizar la fecha y la hora sobre la pantalla LCD

### Esquema

Se muestra en la figura 9. La pantalla LCD se gestiona mediante interface de 4 bits. Las líneas RB0:RB3 se conectan con las líneas D4:D7 para la transferencia de datos. Las líneas RA1, RA2 y RA3 se conectan con E, R/W y RS respectivamente y se emplean como señales de control. Por su parte el módulo receptor GPS simplemente se alimenta desde +Vcc y GND del laboratorio USB-PIC'School. La salida de datos por la señal TTL TX se conecta con la entrada de datos serie RC0.

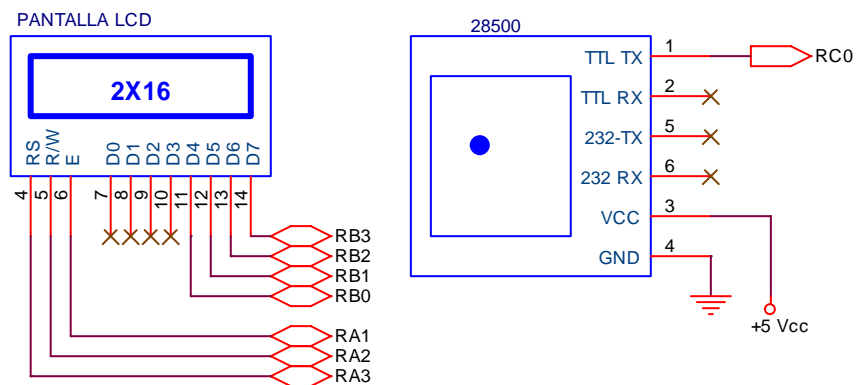


Figura 9. Conexiones para el ejemplo 1

### Comentarios

El ejemplo obtiene la fecha y la hora UTC procedente del receptor GPS 28500 y la visualiza sobre la pantalla LCD del laboratorio. Recordemos que la hora UTC representa el Tiempo Universal Coordinado. Por ello es posible que haya un desfase de 1 hora o más respecto al horario local. En la península este desfase se corresponde con un retraso de 1 o 2 horas en función del horario de invierno o de verano.

Se hace uso de las siguientes funciones:

- RxGPS: Para recibir las tramas GGA y RMC
- GPS\_Date: Para decodificar el campo de la fecha dentro de la trama RMC
- GPS\_Time: Para decodificar el campo de la hora dentro de la trama GGA.

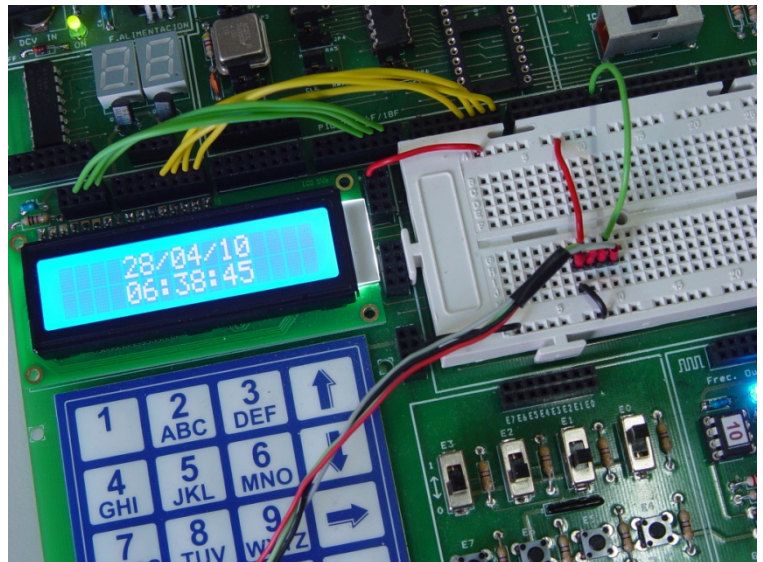


Figura 10. Ejecución del ejemplo 1

En la fotografía de la figura 10 se muestra la ejecución de éste ejemplo. Se puede apreciar que la conexión con el GPS (que está en el exterior) se realiza mediante la manguera de extensión de la que se habló en el apartado 6.3.

## 8.2 Ejemplo 2: Señal, satélites y altura

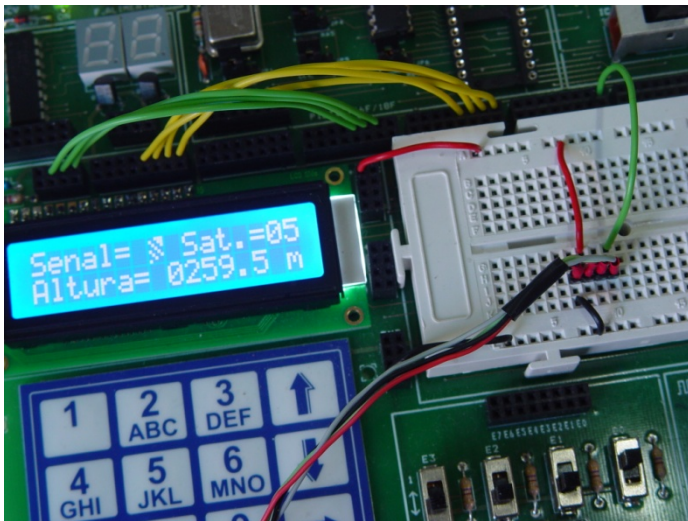
### Objetivos

En este ejemplo se trata de averiguar la calidad de la señal, si se recibe o no algún satélite y el número de estos si son visibles. También se visualiza la altura sobre el nivel del mar.

### Esquema

Se emplea el mismo esquema de conexiones que en el ejemplo anterior

### Comentarios



El ejemplo visualiza sobre el LCD un símbolo que representa si la señal es o no válida, el nº de satélites visibles y la altura en metros sobre el nivel del mar. Se hace uso de las siguientes funciones:

- RxGPS: Para recibir las tramas GGA y RMC.
- GPS\_Signal: Para decodificar el campo que indica la calidad de señal dentro de la trama GGA.
- GPS\_Sat: Para decodificar el campo que indica el número de satélites visibles dentro de la trama GGA
- GPS\_Alt: Para decodificar el campo de la altura dentro de la trama GGA.

En la figura 11 se muestra el resultado de la ejecución de este ejemplo.

Figura 11. Ejecución del ejemplo 2

## 8.3 Ejemplo 3: Visualización de la velocidad

### Objetivos

Obtener y visualizar la velocidad

### Esquema

Se emplea el mismo esquema de conexiones que en el ejemplo anterior

### Comentarios

El ejemplo visualiza sobre el LCD la velocidad. Esta se representa tanto en nudos como en Km/h. Se hace uso de las siguientes funciones:

- RxGPS: Para recibir las tramas GGA y RMC.
- GPS\_Vel\_N: Para decodificar el campo que indica la velocidad en nudos dentro de la trama RMC.
- GPS\_Vel\_K: Para decodificar el campo que indica la velocidad en nudos dentro de la trama RMC y luego calcula el equivalente en Km/h

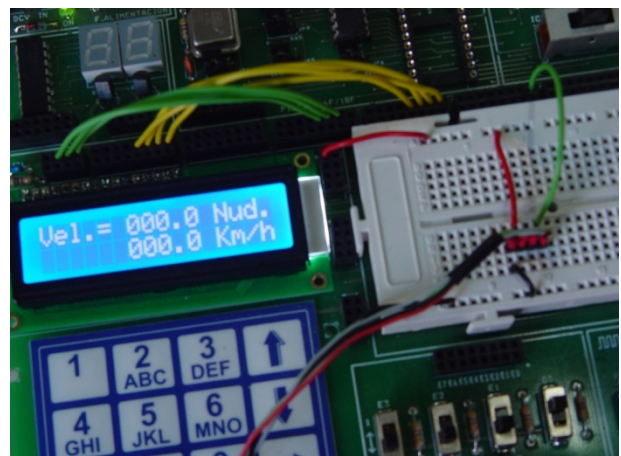


Figura 12. Ejecución del ejemplo 3

Para calcular la velocidad en Km/h se multiplican los nudos por la constante 1.85. En la figura 12 se muestra la ejecución del ejemplo 3. Se aprecia una velocidad de 0 Km/h ya que todos los ejemplos y pruebas se realizan desde una localización fija (la oficina de MSE).

#### 8.4 Ejemplo 4: Obtención de latitud y longitud

##### Objetivos

Capturar y visualizar en formato decimal la latitud y longitud que proporciona el receptor GPS.

##### Esquema

Se emplea el mismo esquema de conexiones que en el ejemplo anterior

##### Comentarios

El ejemplo visualiza sobre el LCD la latitud y longitud expresadas en decimal tal y como se pueden introducir en aplicaciones de mapas (p.e. Google Earth).

Se hace uso de las siguientes funciones:

- RxGPS: Para recibir las tramas GGA y RMC.
- GPS\_Lat\_O: Para obtener y determinar el signo de la orientación en latitud: + = Norte y - =Sur.
- GPS\_Lat\_D: Para obtener la coordenada de latitud en la que los segundos se expresan en decimal, siendo el formato gg mm.ssss compatible con aplicaciones del tipo Google Earth
- GPS\_Lon\_O: Para obtener y determinar el signo de la orientación en longitud: + = Este y - =Oeste.
- GPS\_Lon\_D: Para obtener la coordenada de longitud en la que los segundos se expresan en decimal, siendo el formato ggg mm.ssss compatible con aplicaciones del tipo Google Earth.

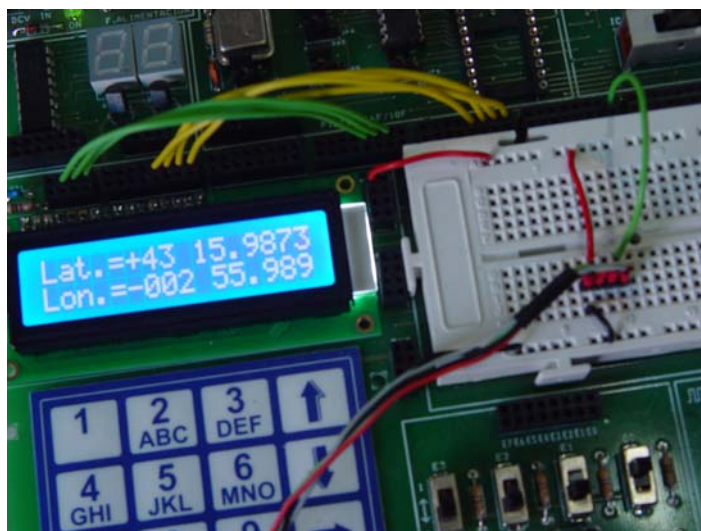


Figura 13. Ubicación próxima a MSE

En la figura 13 se puede apreciar el resultado de la ejecución del presente ejemplo. Las coordenadas se indican se corresponden con las de un establecimiento hostelero próximo a MSE, al que acudimos para hacer este experimento.

En la figura 14 se muestra una captura del programa de mapas de libre distribución Google Earth. La versión demos de este programa se puede descargar desde <http://earth.google.es/download-earth.html>.

A la izquierda de la imagen, en el campo "Volar a" y destacado mediante una elipse, se teclean las coordenadas de latitud y longitud separadas por ',' tal y como nos aparecieron en la pantalla LCD de la figura 13: +43 15.9873, -002 55.989. Inmediatamente el programa nos presenta una imagen de satélite del lugar al que corresponden esas coordenadas (círculo amarillo). En este corresponde a un restaurante próximo a Barrainkua con Alameda de Rekalde en las cercanías de MSE.

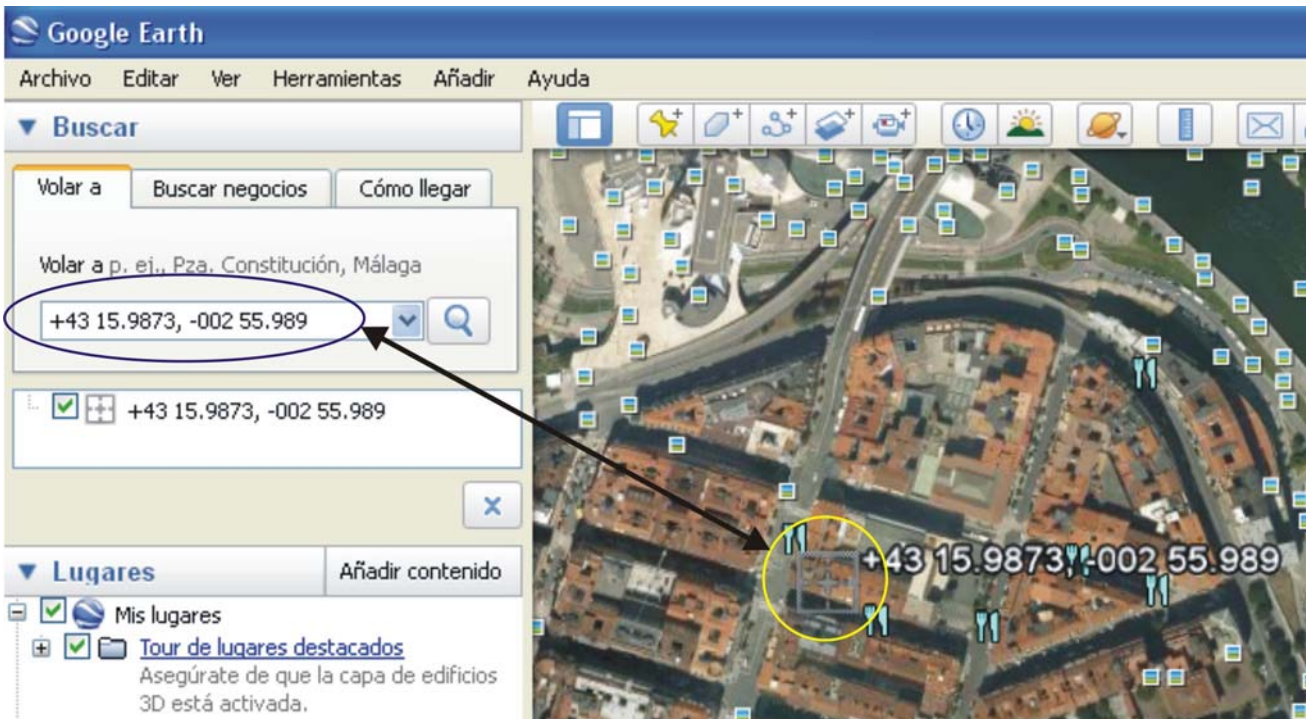


Figura 14. Imagen de satélite de las coordenadas capturadas en el ejemplo 4

## 8.5 Ejemplo 5: Obtención de latitud y logitud

### Objetivos

Capturar y visualizar en formato sexagesimal la latitud y longitud que proporciona el receptor GPS.

### Esquema

Se emplea el mismo esquema de conexiones que en el ejemplo anterior

### Comentarios

El ejemplo visualiza sobre el LCD la latitud y longitud expresadas en sexagesimal con formato  $gg^{\circ}mm'xx.x''$

Se hace uso de las siguientes funciones:

- RxGPS: Para recibir las tramas GGA y RMC.
- GPS\_Lat\_O: Para obtener y determinar el signo de la orientación en latitud: N = Norte y S = Sur.
- GPS\_Lat\_S: Para obtener la coordenada de latitud en la que los segundos se expresan en sexagesimal.
- GPS\_Lon\_O: Para obtener y determinar el signo de la orientación en longitud: E = Este y W = Oeste.
- GPS\_Lon\_S: Para obtener la coordenada de longitud en la que los segundos se expresan en sexagesimal.

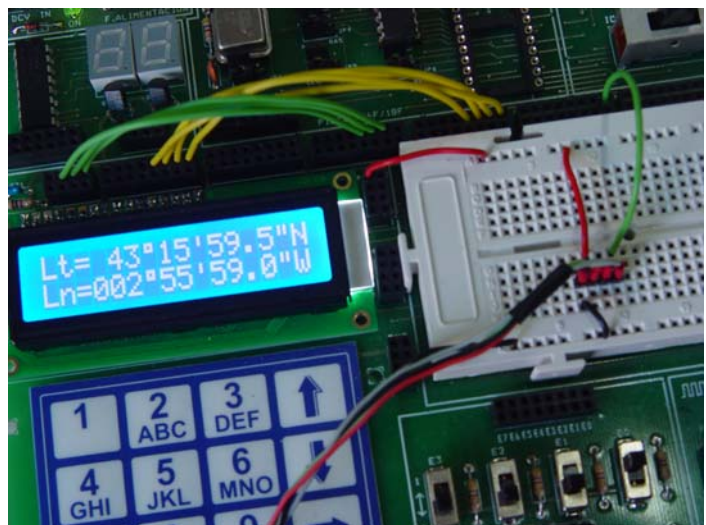


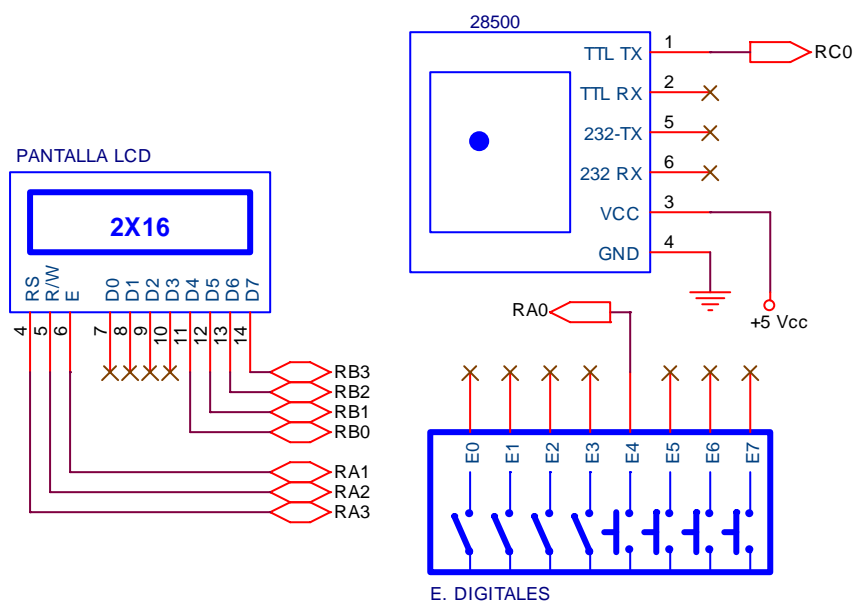
Figura 15. Ejecución del ejemplo 5

## 8.6 Ejemplo 6: Visualización de todos los parámetros típicos de un receptor GPS

### Objetivos

Tratar de hacer un ejemplo de aplicación en el que sobre la pantalla LCD se van visualizando secuencialmente los parámetros más representativos de un receptor GPS comercial: Calidad de la señal, Nº de satélites visualizados, altura sobre el nivel del mar, coordenadas tanto en decimal como en sexagesimal, velocidad, fecha y hora UTC

### Esquema



Se muestra en la figura 16. Es muy similar a los esquemas empleados en todos los ejemplos anteriores. En esta ocasión empleamos el pulsador E4 conectado con la línea de entrada RA0 para seleccionar secuencialmente el modo de visualización

Figura 16. Esquema del ejemplo 6

### Comentarios

Consiste en un ejemplo que trata de aglutinar a todos los anteriores así como emplear todas las funciones de la librería GPS28500\_PIC1.INC. Se visualiza sobre el LCD todos los parámetros propios de un receptor GPS con la siguiente secuencia:

- 1.- Calidad de la señal, nº de Satélites visibles y altura en m. sobre el nivel del mar
- 2.- Latitud y longitud en formato decimal compatible con Google Earth
- 3.- Latitud y longitud en formato sexagesimal similar a otros dispositivos GPS
- 4.- Velocidad expresada tanto en nudos como en Km/h
- 5.- Fecha y hora UTC (Tiempo Universal Coordinado)

Mediante el pulsador E4 conectado en la entrada RA0 se va modificando esa secuencia cada vez que se acciona.

## 9. PROYECTO FINAL

Hasta ahora, en todos los ejemplos propuestos, hemos empleado el receptor GPS 28500 junto con el laboratorio PIC'School como plataforma de prueba y experimentación para el desarrollo y puesta a punto de los diferentes programas. Sin embargo debemos reconocer que una aplicación real basada en GPS es esencialmente una aplicación portátil en la que debemos realizar un último paso.

Este consiste en realizar un equipo a medida, pequeño, económico y autónomo que pueda ser utilizado en cualquier lugar: en la calle, en el monte, en la playa, en vehículos, etc... Basándonos pues en el ejemplo 6 vamos a desarrollar un prototipo electrónico que reúna esas condiciones tal y como se muestra en la figura 17.

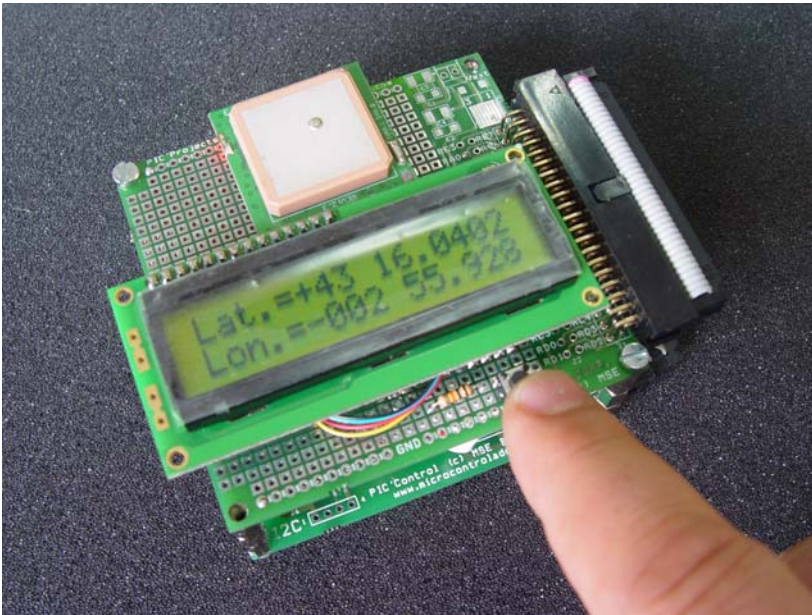


Figura 17. Prototipo de un receptor GPS totalmente autónomo.

## 9.1 Esquema eléctrico

Todo el conjunto se basa en la tarjeta de control de propósito general PIC'Control, económica y de reducida dimensiones así como en la tarjeta para el montaje de prototipos PIC'Project, tal y como se muestra en la figura 18.

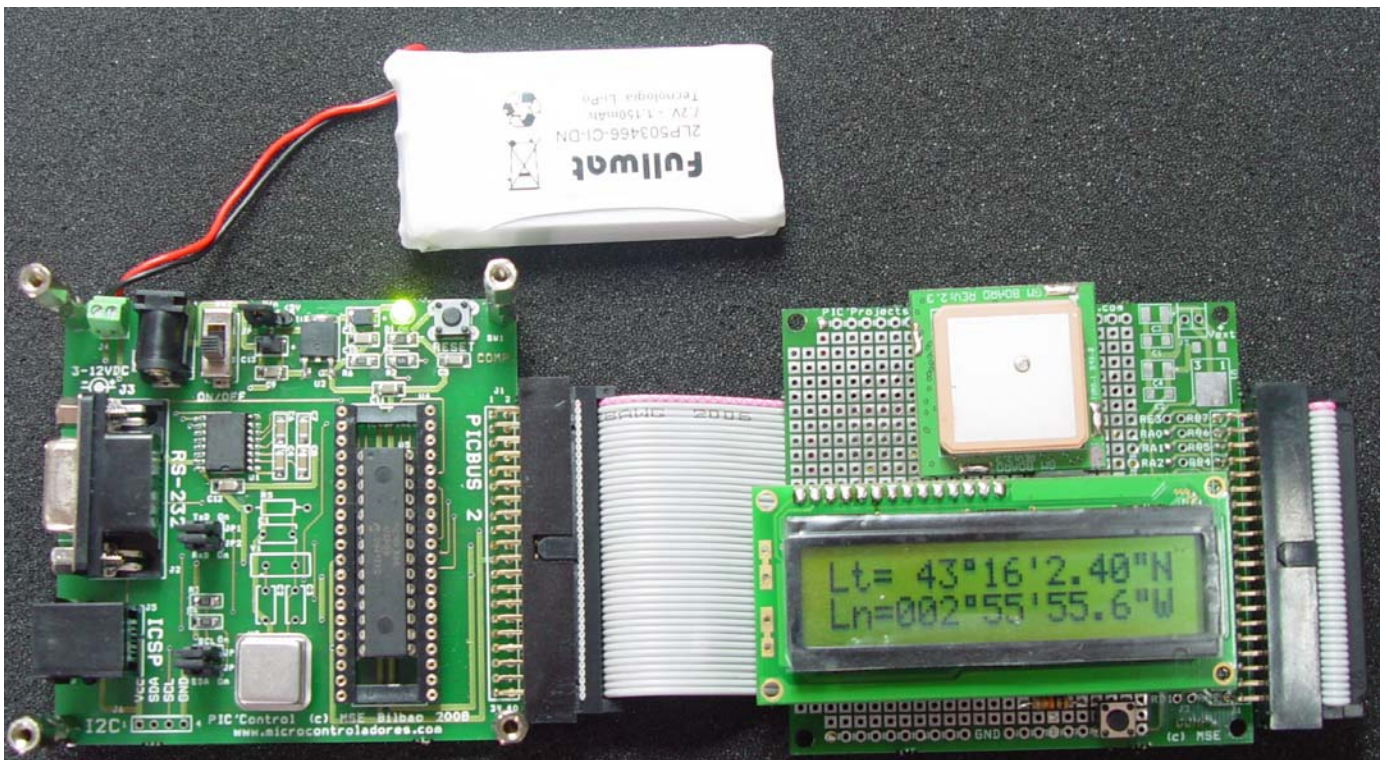
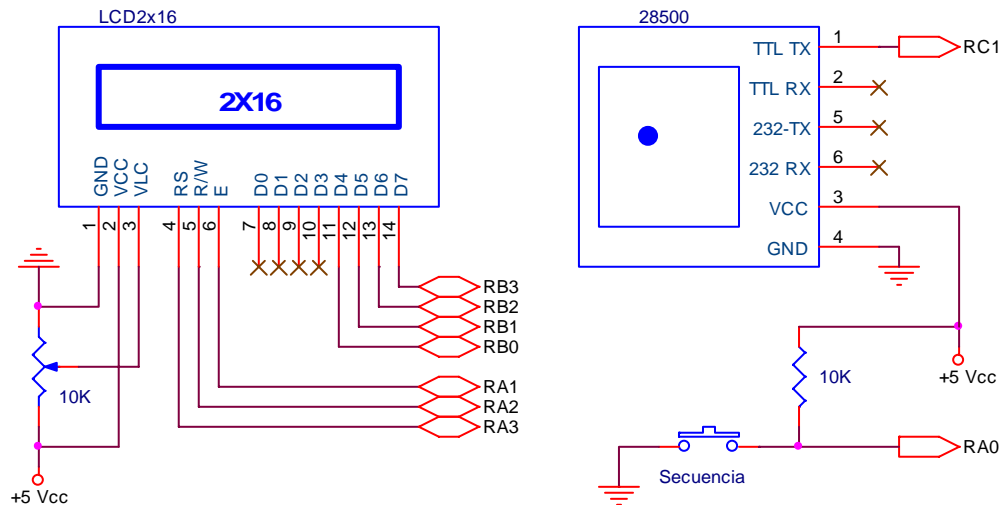


Figura 18. Las tarjetas PIC'Control y PIC'Project

Ambas están diseñadas, fabricadas y comercializadas por Ingeniería de Microsistemas Programados. Sobre PIC'Project se realiza el montaje con todos los elementos según el esquema de la figura 19.



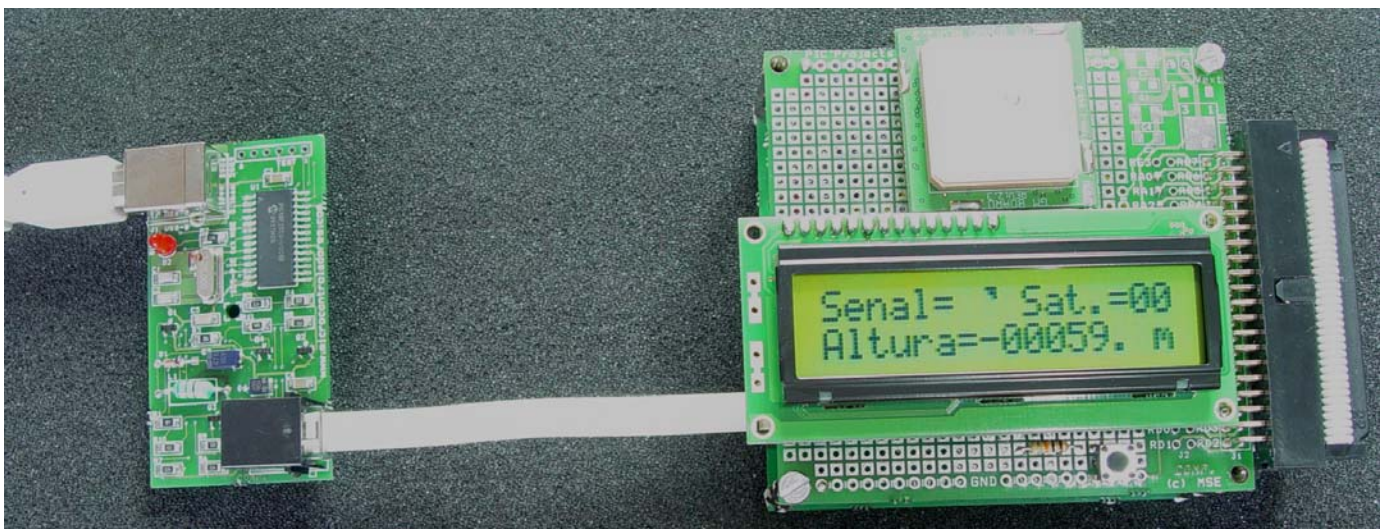
**Figura 19.** Montaje a realizar sobre la tarjeta PIC'Project

La pantalla LCD se conecta como lo hemos venido haciendo en los ejemplos anteriores. Lógicamente se le añade la alimentación y el potenciómetro de 10K para el control del contraste. El módulo receptor GPS 28500 se conecta como siempre: la alimentación y la salida de datos TTL TX que se conecta con la línea RC1. Por último montamos el pulsador junto con la resistencia pull-up de 10K que se conecta con la entrada RA0 y gracias al cual podemos seleccionar secuencialmente la visualización deseada según el ejemplo 6.

## 9.2 Depuración/grabación

Se supone que el programa de aplicación es el correspondiente al ejemplo 6 que ya se probó, depuró y grabó con ayuda del laboratorio USB-PIC'School. Basta con retirar el PIC16F886 recién grabado e insertarlo sobre la tarjeta de control PIC'Control.

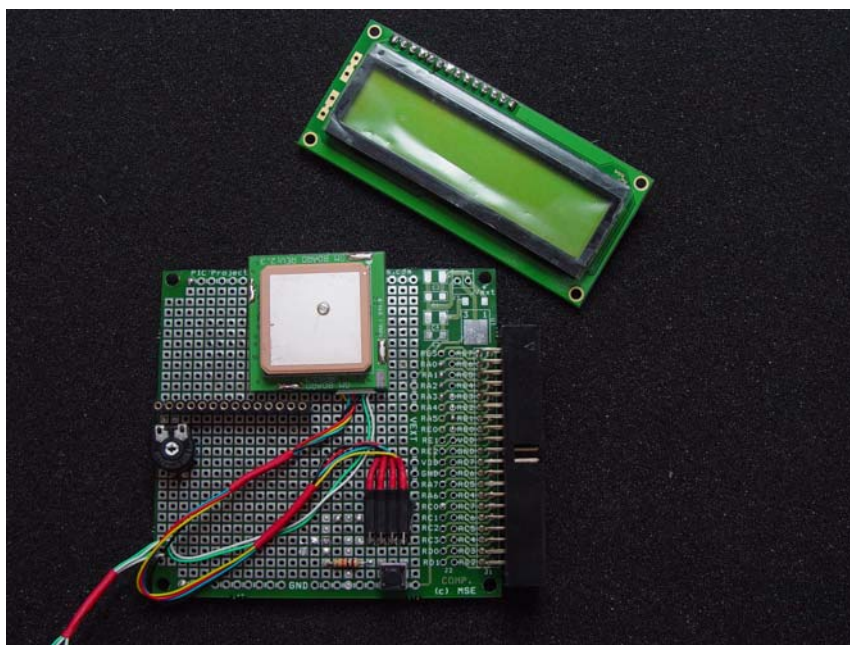
Otra alternativa es el empleo de nuestro Depurador/grabador ICD-PIC. Podemos tener insertado de forma permanente, sobre la tarjeta PIC'Control, el controlador PIC16F886. El ICD-PIC lo emplearemos para grabar el programa de aplicación. En última instancia, gracias a las capacidades de depuración de ICD-PIC, también podremos hacer todo tipo de modificaciones, ajustes de última hora y trabajos de campo sin tener que pasar de nuevo por el laboratorio USB-PIC'School. La figura 20 muestra la conexión de todo nuestro prototipo con el depurador/grabador ICD-PIC.



**Figura 20.** Depuración/grabación desde el ICD-PIC

### 9.3 Listado de componentes

Todos los componentes empleados en este proyecto son de fácil localización parte de los cuales se aprecian en la figura 21.



**Figura 21.** Componentes del proyecto

Se pueden adquirir directamente desde la tienda web de Ingeniería de Microsistemas Programados S.L. En el siguiente listado se presentan los componentes empleados así como su código o referencia.

CANTIDAD	REFERENCIA	DESCRIPCION
1	PIC'Control	Tarjeta de control de propósito general con cristal de cuarzo de 4MHz
1	PIC'Project	Tarjeta para montaje de prototipos. Incluye cable plano para el interface con PIC'Control
1	PIC16F886	Microcontrolador
1	P28500	Receptor GPS
1	06LCD2X16	Pantalla LCD de 2 x 16 caracteres
1	34PT10V103	Potenciómetro de 10K
1	30R1C103	Resistencia de carbón de 10K
1	61KSA	Pulsador c. impreso
1	54TIRA361D	Tira de postes macho para conexión del LCD y del receptor GPS
1	54BASE361PT7	Tira de conectores hembra para conexión del LCD y del receptor GPS
1	CABLE	Rollo de cable rígido de 10m y 0.6mm de grosor
1	CC3015X	Rollo de 15m de cable de rapinar 30AWG (0.25mm)
1	ESTANOTUBO	Rollo de estaño en tubo (20gr)
1	APB2LP503466	Batería LIPO de 7.4V / 1150 mAh (opcional)
1	FU-CLI100	Cargador de baterías LIP (opcional)
1	FU-ADPV-60A-12	Alimentador de salida fija a 12V/1A (opcional)