

Diseño y Desarrollo de un Módulo para el Almacenamiento de Coordenadas GPS mediante el  
uso de un Micro Controlador

Israel Emmanuell Pazmiño Tamayo

Universidad Internacional SEK

Nota de Autor

Israel Emmanuell Pazmiño Tamayo, Facultad de Ingeniería Mecánica, Universidad  
Internacional SEK; Director Gustavo Moreno.

Cualquier correspondencia concerniente a este trabajo puede dirigirse a:  
[profesor\\_epn@yahoo.es](mailto:profesor_epn@yahoo.es).

**Declaración Juramentada**

Yo, ISRAEL EMMANUELL PAZMIÑO TAMAYO, con cédula de identidad 172144312-3, declaro bajo juramento que el trabajo aquí desarrollado es de mi autoría, que no ha sido previamente presentado para ningún grado o calificación profesional; y, que se ha consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración, cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la UNIVERSIDAD INTERNACIONAL SEK, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normativa institucional vigente.

ISRAEL EMMANUELL PAZMIÑO TAMAYO

C.I.: 1721443123

## Índice de contenido

Declaración Juramentada .....	2
Índice de Contenidos .....	3
Índice de Tablas y Figuras .....	6
Resumen .....	9
Abstract .....	10
Introducción .....	11
Estado del Arte .....	11
Método .....	16
Componentes del Módulo de Almacenamiento de Coordenadas GPS .....	16
Módulo Arduino Uno .....	16
Módulo GPS .....	17
Módulo de almacenamiento SD.....	18
Botones de selección.....	19
Alertas visuales .....	20
Funcionamiento del Módulo de Almacenamiento de Coordenadas GPS .....	21
Ensamblaje del Módulo de Almacenamiento de Coordenadas GPS .....	22
Diseño del software .....	26
Procesamiento de datos en el programa GPS Visualizer y Google Earth .....	29
Experimentos .....	32

Experimento 1.....	32
Experimento 2.....	32
Resultados .....	33
Experimento 1.....	33
Experimento 2.....	36
Recorrido 1 .....	36
Recorrido 2 .....	38
Recorrido 3 .....	41
Discusión .....	45
Experimento 1.....	45
Experimento 2.....	46
Recorrido 1 .....	46
Recorrido 2 .....	47
Recorrido 3 .....	48
Conclusiones.....	49
Recomendaciones .....	49
Referencias Bibliográficas .....	51
Anexo A .....	54
Anexo B .....	55
Anexo C .....	56
Anexo D .....	67

Anexo E .....	69
---------------	----

Anexo F .....	71
---------------	----

Anexo G .....	73
---------------	----

## Índice de Tablas y Figuras

### Tablas

Tabla 1. Protocolo NMEA RMC .....	18
Tabla 2. Resultados de muestras GPS sobre el mismo punto .....	33
Tabla 3. Distancia entre el punto de origen y cada dato GPS .....	35
Tabla 4. Resultados de muestras GPS del primer recorrido.....	36
Tabla 5. Resultados de muestras GPS del segundo recorrido .....	39
Tabla 6. Resultados de muestras GPS del tercer recorrido .....	42

### Figuras

Figura 1. Partes del Sistema de Información Geográfica .....	12
Figura 2. Módulo Arduino Uno .....	16
Figura 3. Módulo GPS Dexter Industries .....	17
Figura 4. Módulo de almacenamiento SD .....	19
Figura 5. Botones de selección .....	20
Figura 6. LEDs indicadores de estado .....	20
Figura 7. Diagrama de bloques del funcionamiento del módulo GPS .....	21
Figura 8. Placas de acrílico de diferente tamaño .....	22
Figura 9. Montaje módulo Arduino y GPS .....	23
Figura 10. Ubicación del módulo SD y batería de alimentación .....	23

Figura 11. Ubicación del módulo Arduino, GPS, SD y batería .....	24
Figura 12. Ubicación de botones, indicadores LED y tornillos sobre la tapa superior .....	24
Figura 13. Tapas laterales .....	25
Figura 14. Dimensiones de la caja de acrílico .....	26
Figura 15. Interfaz del software Arduino 1.0.4. ....	27
Figura 16. Diagrama de flujo del módulo de almacenamiento de coordenadas GPS .....	28
Figura 17. Ventana principal GPS Visualizer .....	29
Figura 18. Menú de configuración GPS Visualizer .....	30
Figura 19. Ventana principal del programa Google Earth .....	31
Figura 20. Ejemplo de una ruta trazada en Google Earth .....	31
Figura 21. Datos GPS Magellan 315.....	34
Figura 22. Datos del Módulo GPS .....	34
Figura 23. Combinación de datos GPS y marca de origen.....	35
Figura 24. Gráfica de los datos de muestra del GPS Magellan 315.....	37
Figura 25. Gráfica de los datos de muestra del Módulo GPS .....	37
Figura 26. Gráfica de los datos de muestra del GPS Magellan 315 y todos los datos del Módulo GPS .....	38
Figura 27. Gráfica de los datos de muestra del GPS Magellan 315.....	40
Figura 28. Gráfica de los datos de muestra del Módulo GPS .....	40

Figura 29. Gráfica de los datos de muestra del GPS Magellan 315 y todos los datos del Módulo GPS .....	41
Figura 30. Gráfica de los datos de muestra del GPS Magellan 315 .....	43
Figura 31. Gráfica de los datos de muestra del Módulo GPS .....	43
Figura 32. Gráfica de los datos de muestra del GPS Magellan 315 y todos los datos del Módulo GPS .....	44



### Resumen

Se presenta el diseño y construcción de un módulo de almacenamiento de coordenadas GPS en una memoria SD funciona con un hardware y software libre con una interfaz de comunicación amigable con el usuario.

El GPS del módulo permite captar las señales de los satélites de posicionamiento geográfico basado en el protocolo NMEA GPRMC, del cual se obtienen datos de latitud y longitud y que son almacenadas en una memoria SD en formato .txt para su posterior análisis.

Para visualizar la ruta previamente establecida, es necesario ingresar los datos almacenados en la memoria SD en el computador, descargarlos y convertirlos en un archivo KML con ayuda del programa GPS Visualizer para su posterior uso en Google Earth. Este último programa permite visualizar los datos obtenidos con el GPS y compararlos con la ruta establecida originalmente.

**Palabras Clave:** Sistema de Información Geográfica, Sistema de Posicionamiento Global, Arduino, Microcontrolador, Tarjeta SD.

### **Abstract**

The GPS module and an SD memory works with free hardware and free software, with friendly communication with the user.

GPS module can obtain signals from geographic positioning satellites based on NMEA GPRMC protocol, the latitude and longitude are obtained from GPS and that are stored on an SD memory .txt file format for further analysis.

To display the previously established route, the data stored on the SD card must download and convert them to a KML file format using GPS Visualizer program for to use in Google Earth. Google Earth program allows visualizing data obtained from the GPS module and comparing them with the originally established route.

**Key Words:** Geographic Information System, Global Position System, Arduino, Microcontroller, SD card.

## **Introducción**

En la actualidad la tecnología y los Smart Phones está copando los mercados con programas que facilitan de una manera sencilla la ubicación mediante el uso de un GPS, pero no es factible recopilar los datos que se obtienen del mismo debido a su limitada capacidad para guardarlos y analizarlos por separado. Además, los datos obtenidos ya sea por un dispositivo GPS o un Smart Phone no permiten la integración de un sistema que combine hardware, software y datos geográficos que proporcionen un uso para beneficio de la comunidad dando un libre acceso a los mismos (Open Source).

A pesar de que existen dispositivos que permiten receptar datos de ubicación GPS; el desarrollo de un hardware y software que permita almacenar estos datos no ha sido implementado de tal manera que sea accesible para personas o empresas que deseen desarrollar un Sistema de Información Geográfica (SIG o GIS) que beneficie a la comunidad con el afán de establecer un sistema económico social, solidario y sostenible. (SENPLADES, 2013)

Por este motivo se evidencia la necesidad de diseñar un sistema abierto tanto en hardware como en software, que permita a los usuarios realizar cambios para solventar problemas específicos de cada problema.

## **Estado del Arte**

Un Sistema de Información Geográfica, como se puede observar en la Figura 1, es un conjunto de componentes que están ordenadamente relacionados para analizar, procesar y obtener datos que contienen una información espacial. Sus principales componentes son: hardware, software, datos geográficos y herramientas de análisis.

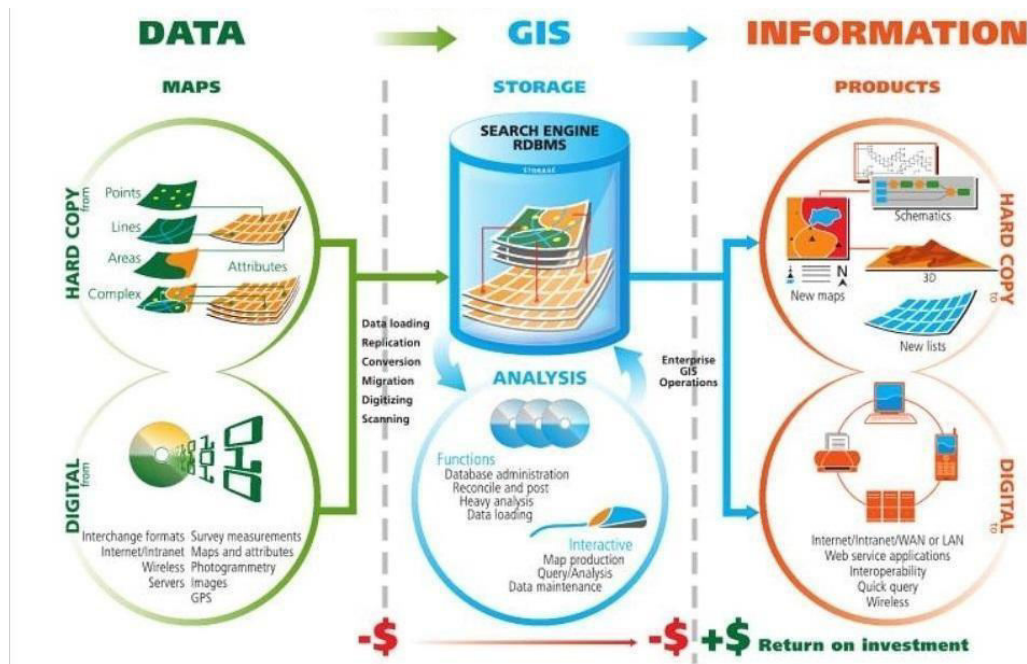


Figura 1. Partes del Sistema de Información Geográfica

TOMLINSON, R. (2007). *Thinking about GIS* (Third Edition). New York – USA: Ingram Publisher Services.

(Pg. 2)

Particularmente estos componentes para los GIS, están diseñados para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y gestión geográfica (TOMLINSON, 2007).

Para la implementación de un sistema GIS, es necesario el uso de un micro controlador que permita procesar datos. Valdés y Pallas lo describen como: "Un micro controlador es un circuito integrado que permite ser programado según se lo requiera y que es capaz de ejecutar las órdenes grabadas en su memoria según se lo establezca en su programación" (Valdés, Pallás, 2007, p. 13). Un microcontrolador combina los recursos fundamentales disponibles en un microcomputador, es decir, la unidad central de procesamiento (CPU), la memoria y los recursos de entrada y salida, en un único circuito integrado.

El GIS requiere de datos obtenidos de los satélites que rodean la tierra y que proporcionan la información necesaria de la ubicación. El GPS proporciona la localización continua y la información de tiempo, en cualquier lugar del mundo. El GPS es un sistema de un solo sentido (pasivo), es decir, los usuarios sólo pueden recibir las señales del satélite.

El GPS consta, nominalmente, de una constelación de 24 satélites operacionales. Esta constelación, conocida como la capacidad operativa inicial (IOC), se completó en julio de 1993 (El-Rabbany, 2002). Para asegurar una cobertura mundial continua, los satélites GPS están dispuestos de modo que cuatro satélites se colocan en cada uno de los seis planos orbitales. Con esta geometría de la constelación de satélites GPS, de cuatro a diez serán visibles en cualquier lugar en el mundo y sólo se necesitan cuatro satélites para proporcionar el posicionamiento o la ubicación.

Una vez recabados los datos, es necesario almacenarlos en un dispositivo que permita su fácil acceso y de igual manera su interacción con otros equipos. El almacenamiento de estado sólido (también llamado memoria flash) es una tecnología que guarda los datos en circuitos borrables regrabables. Se utiliza mucho en dispositivos portátiles como cámaras digitales, reproductoras de música MP3, computadoras laptop y teléfonos celulares.

Existen varios tipos de almacenamiento de estado sólido para los consumidores actuales. Ibrahim (2010, p. 107) determina que: “los formatos para estas pequeñas tarjetas planas son: Compact Flash, Multi Media, Secure Digital (SD) y Smart Media”.

Las tarjetas SD son, probablemente, las tarjetas de memoria más utilizadas hoy en día. La tarjeta SD fue desarrollada originalmente por Matsushita, SanDisk y Toshiba en el año 2000 (Ibrahim, 2010). Las tarjetas SD estándar están disponibles con capacidades de 4 MB a 4 GB. Recientemente se creó un nuevo tipo de tarjeta SD, se le llama la tarjeta SD de alta capacidad (SDHC), se ha desarrollado con una capacidad de 4 a 32 GB.

Toda la integración de hardware como el módulo Arduino, el GPS y SD shields, y el software de programación de Arduino, permiten que el SIG funcione simultáneamente para poder obtener los datos que posteriormente pueden ser analizados y procesados.

En la actualidad existen una innumerable cantidad de dispositivos GPS que permiten la ubicación de las personas mediante el uso de satélites en ciudades, carreteras y en lugares de difícil acceso; sin embargo, estos dispositivos son costosos y complicados de manejar en su mayoría.

Un claro ejemplo de los dispositivos disponibles son los Smart Phones. Estos dispositivos tienen un sistema integrado de GPS que permite guiarse en cualquier parte del mundo e inclusive realizar el trazo de rutas para la movilización. El principal inconveniente es que solo permiten realizar trazos de rutas o dar la ubicación y no permite recabar los datos conseguidos para compartirlos con otros dispositivos o para su procesamiento de forma particular para la elaboración de un GIS.

Otra forma de obtener la ubicación geográfica es mediante el uso de los GPS comunes que existen en varias marcas y modelos. La mayoría de estos dispositivos son complejos de utilizar y su costo varía según las funciones y precisión de los mismos. Uno de los inconvenientes con estos dispositivos es que sus funciones son pre programadas y el usuario se debe ajustar a las mismas sin tener el poder de modificarlas a su conveniencia.

Por otra parte, el uso de un software y hardware libre permite manipular de mejor manera los datos obtenidos y modificarlos según los requerimientos que el usuario desee, de esta forma es notablemente más simple la manera de realizar el levantamiento de un GIS.

A pesar de todas las facilidades que dan los dispositivos de ubicación GPS y los Smart Phones, siempre mantienen las mismas características en las cuales no se puede realizar cambios estructurales o de programación. Esta situación mantiene un mercado concentrado en solo prestar un servicio de ubicación en el que no se permite el libre intercambio de datos para

su uso o proceso de la manera más conveniente al usuario o a su vez dar una prestación diferente a solo la ubicación en un punto geográfico.

Es por ello que se diseña un prototipo de obtención de datos de ubicación de bajo costo que permite integrar los sistemas de hardware, software y datos geográficos en un sistema SIG. El prototipo ha sido creado con el afán de prestar mejores servicios a la comunidad, obteniendo un sistema que permite la adquisición de datos con un óptimo desempeño y de libre manipulación.

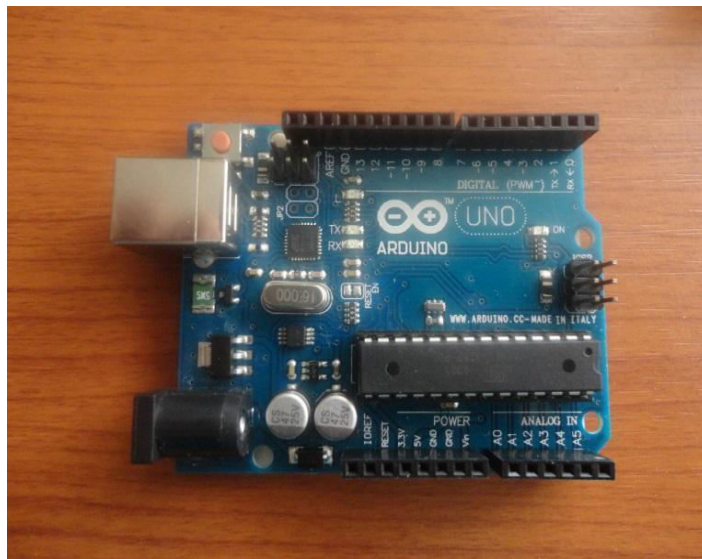
### Método

Para la elaboración del módulo de almacenamiento de coordenadas GPS mediante el uso de un micro controlador, se creó un software que utiliza los datos obtenidos mediante un módulo GPS y los procesa de tal manera que la información es almacenada en una memoria SD en un formato de texto plano “.txt” para ser usado en el programa GPS Visualizer, que a su vez, creará un nuevo archivo KMZ legible en el programa Google Earth.

### Componentes del Módulo de Almacenamiento de Coordenadas GPS

#### Módulo Arduino Uno.

El módulo Arduino Uno es una placa electrónica basada en el microprocesador Atmega328 (ARDUINO. 2006). Cuenta con 14 pines digitales de entrada / salida, de los cuales 6 pueden ser utilizados como salidas PWM y 6 entradas analógicas, un cristal de cuarzo cerámico de 16 MHz, una conexión USB, un conector de alimentación y un botón de reinicio. (Figura 2)



*Figura 2. Módulo Arduino Uno*



Las entradas 3 y 4 son utilizadas para comunicarse con el módulo GPS. Entradas 5, 6, 7 para las alertas visuales; y por último, las entradas 10, 11, 12, 13 para la emisión y recepción de datos hacia el módulo SD. El diagrama de conexión eléctrica se muestra en el Anexo A

### **Módulo GPS.**

El GPS que proporciona los datos al módulo Arduino es de Dexter Industries (Figura 3), el cual los recepta en base al protocolo National Marine Electronics Association (NMEA). Este protocolo consta de varios mensajes que aportan diferentes datos y entre los cuales están: GGA, GLL, GSA, GSV, MSS, RMC, VTG, ZDA y 150. (NMEA Reference Manual, 2013).



*Figura 3. Módulo GPS Dexter Industries*

El protocolo utilizado es el Recommended Minimum Specific GNSS Data (RMC). Los datos de este protocolo se explican en la tabla 1.

Tabla 1

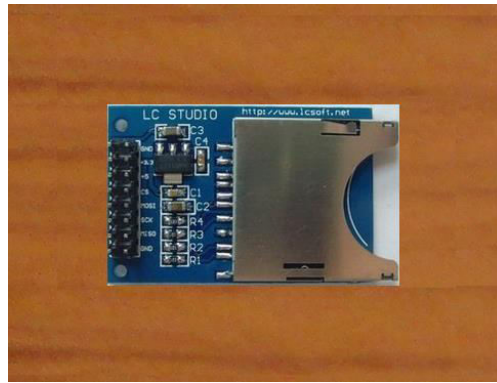
*Protocolo NMEA RMC*

<b>Protocolo NMEA</b>			
\$GPRMC,161229.487,A,3723.2475,N,12158.3416,W,0.13,309.62,120598,*,*10			
<b>Nombre</b>	<b>Ejemplo</b>	<b>Unidad</b>	<b>Descripción</b>
Nombre del Mensaje	\$GPRMC		Encabezado del protocolo RMC
Hora UTC	161229.487		hh mm ss.sss
Estado	A		A = dato válido V = dato invalido
Latitud	3723.2475		GG mm.mmmm
Indicador N/S	N		N = norte / S = sur
Longitud	12158.3416		GG mm.mmmm
Indicador E/W	W		E = este / W = oeste
Velocidad respecto tierra	0.13	Nudos	
Rumbo respecto tierra	309.62	Grados	Verdaderos
Fecha	120598		dd mm aa
Variación Magnética		Grados	E = este / W = oeste
Checksum	*10		

Cada uno de los datos de la sentencia del protocolo NMEA está separado por una coma, de los cuales se utilizaron los siguientes: nombre del mensaje, hora UTC, estado, latitud, indicador N/S, longitud, indicador E/W y fecha.

**Módulo de almacenamiento SD.**

El módulo de almacenamiento SD (Figura 4), está acoplado al módulo Arduino y GPS conectado a los terminales 10, 11, 12 y 13.



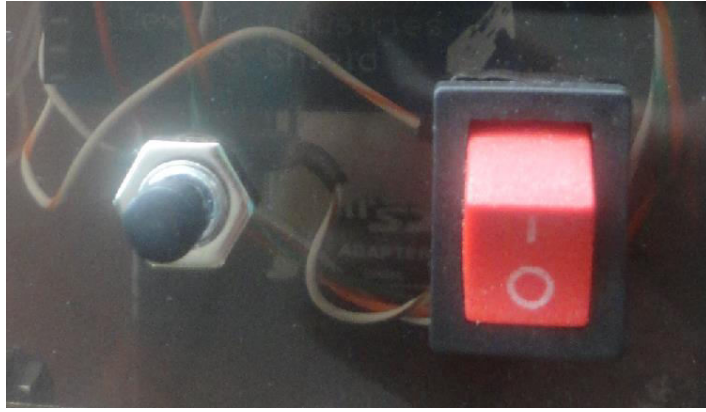
*Figura 4.* Módulo de almacenamiento SD

El módulo SD consta con los siguientes pines que son conectados a los terminales anteriormente indicados:

- CLK (Línea de reloj): Es la señal de reloj (pin 13).
- MOSI (MasterOut-SlaveIn): Línea por donde el maestro envía y el esclavo recibe (pin 11).
- MISO (MasterIn-SlaveOut): Línea por donde el maestro recibe y el esclavo envía (pin 12).
- CS (Chip Select)/ SS (Slave Selector): Conecta o desconecta la operación del dispositivo con el que comunicamos. Permite la comunicación de varios esclavos a un mismo maestro (pin10).

### **Botones de selección.**

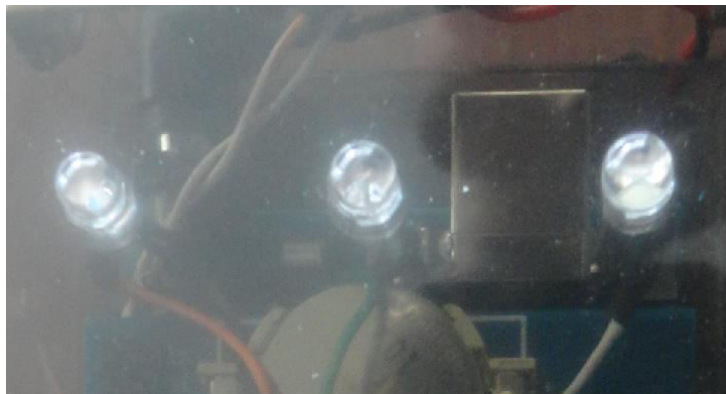
El prototipo consta de 2 botones externos visibles como se observa en la Figura 5. Un botón de encendido y apagado del módulo que permite encender o apagar el mismo cuando no esté operando (derecha). De igual manera se tiene un botón de Reset, el cual cumple la función de reiniciar el programa sin la necesidad de apagar el equipo en su totalidad (izquierda).



*Figura 5. Botones de selección*

### **Alertas visuales.**

El prototipo consta con 3 diodos LED de diferentes colores (rojo, verde y blanco), que permiten indicar el funcionamiento del mismo (Figura 6).



*Figura 6. LEDs indicadores de estado.*

El LED rojo (izquierda), permanece encendido cuando los datos receptados son incorrectos. El LED verde (centro), se enciende y permanece en ese estado cuando el dato obtenido es correcto. Finalmente, el LED blanco (derecha), parpadea con cada dato guardado en la tarjeta de memoria SD.

### Funcionamiento del Módulo de Almacenamiento de Coordenadas GPS

Al momento de encender el módulo GPS, el primer paso es detectar si existe una tarjeta de memoria SD. Realizado este procedimiento, el GPS empieza a recibir los datos de la constelación de satélites que se encuentran alrededor de la tierra. Estos datos deben ser procesados y depurados por el módulo Arduino para luego grabar los datos depurados y entregar una respuesta visual en los LED como se indica en el diagrama de bloques de la figura 7.

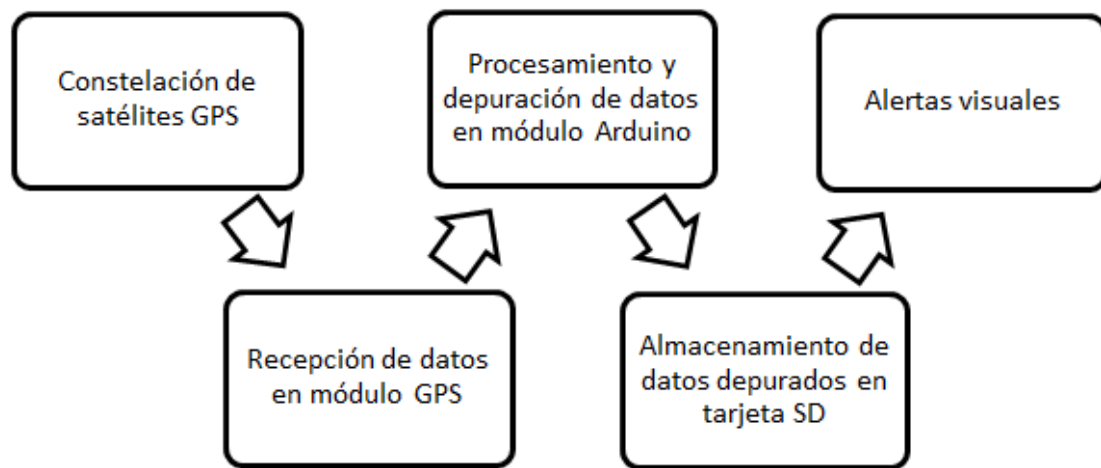


Figura 7. Diagrama de bloques del funcionamiento del módulo GPS

Cada vez que el módulo se encienda, se creará un archivo nuevo en la tarjeta SD en el que se grabarán los datos depurados. De igual manera, cada vez que se presione el botón reset, el programa creará un archivo nuevo.

El proceso de depuración de datos consiste en recolectar del protocolo NMEA sólo los que corresponden a la sentencia RMC que son: Hora, Latitud, Longitud y Fecha, los cuales son almacenados en la tarjeta de memoria SD.

Si la tarjeta de memoria no es insertada desde el inicio y al momento de encender el módulo, se disparará una alerta visual encendiendo los 3 LED y haciéndolos parpadear hasta

que se inserte la tarjeta SD y se reinicie el módulo aplastando el botón de reset o se apague y encienda nuevamente.

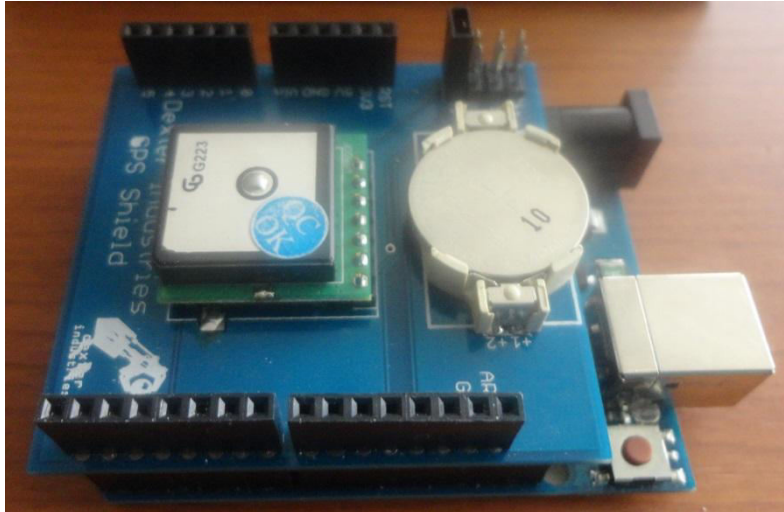
### **Ensamblaje del Módulo de Almacenamiento de Coordenadas GPS**

Es necesario resguardar los equipos de tal manera que estén protegidos contra golpes y sean fáciles de transportar, para lo cual se construyó una caja que contiene los módulos Arduino, GPS, SD y la batería, así como también los indicadores visuales y los botones. La caja está ensamblada principalmente de acrílico de 3mm debido a su resistencia y su fácil moldeo. (Figura 8)



*Figura 8.* Placas de acrílico de diferente tamaño.

Las dimensiones de la caja están basadas en el tamaño y forma de ensamblaje de los módulos, considerando que el módulo Arduino y GPS se apilable uno encima de otro, como se observa en la Figura 9.



*Figura 9. Montaje módulo Arduino y GPS.*

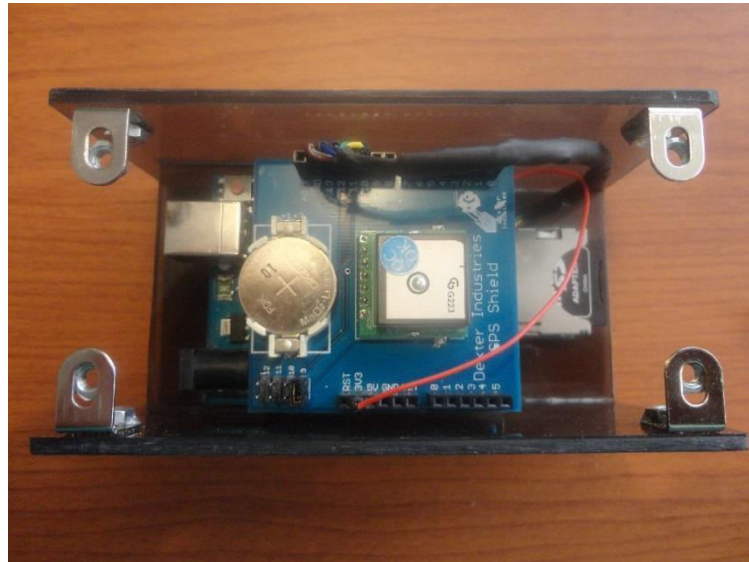
El módulo SD, debido a que no es apilable como el módulo Arduino y GPS, requiere de una disposición diferente y alimentación mediante cables que se dirigen al módulo Arduino, ubicado sobre la base y de tal manera que el acceso a la tarjeta SD quede sobre el costado. La batería de alimentación está ubicada sobre el costado, con 2 separadores sobre los cuales irá montado el módulo Arduino y GPS como se indica en la figura 10.



*Figura 10. Ubicación del módulo SD y batería de alimentación.*

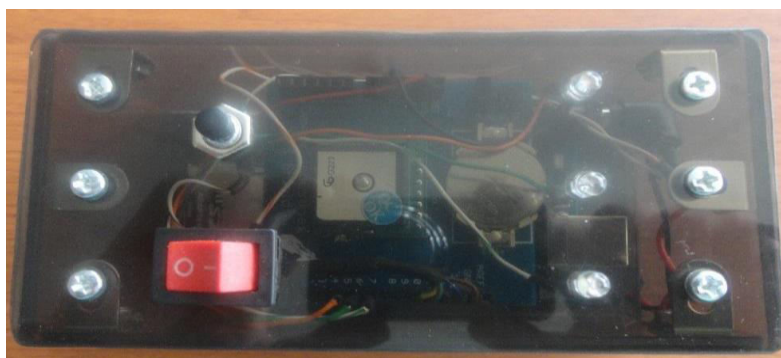


Los módulos Arduino y GPS están ubicados en un nivel superior tomando como referencia la altura de la batería. Estos quedan casi centrados, permitiendo un fácil acceso a la batería y la tarjeta SD evitando posibles desconexiones (Figura 11).



*Figura 11.* Ubicación del módulo Arduino, GPS, SD y batería.

La tapa superior consta con 6 tornillos, 4 sobre las esquinas que sirven para sujetar la misma a la carcasa y 2 centrales que sujetan las tapas laterales. Los botones en la izquierda y los indicadores LED en la derecha, están montados con el espacio suficiente como para manipular cada uno de estos y chequear su funcionamiento o su posible cambio en caso de daño. (Figura 12)



*Figura 12.* Ubicación de botones, indicadores LED y tornillos sobre la tapa superior.

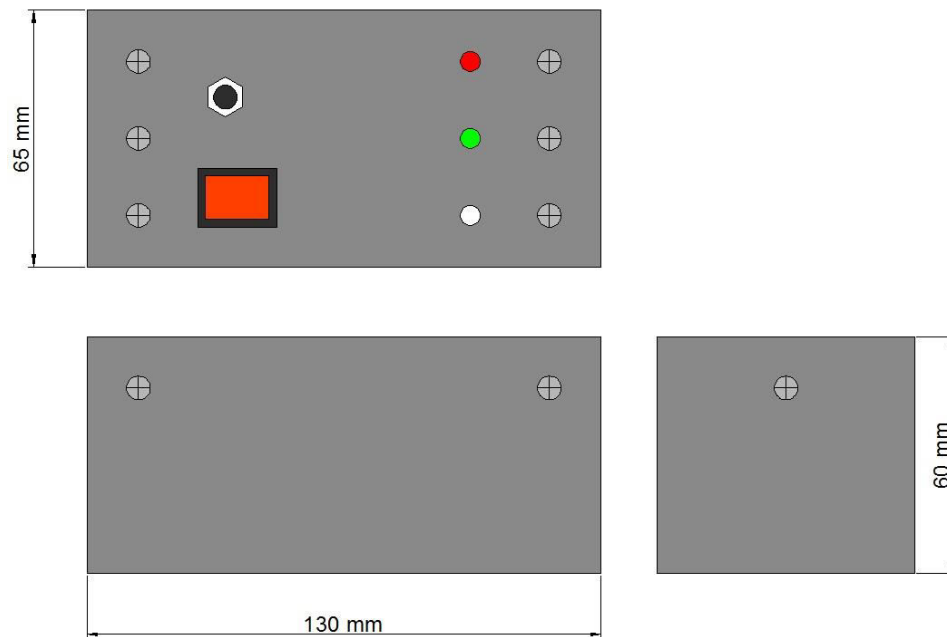


Una particularidad de la caja del módulo, es que cuenta con 2 tapas laterales que son separables para acceso a la batería y de igual manera para la tarjeta SD. Estas tapas se ajustan mediante un tornillo que va emplazado en la tapa superior. (Figura 13)



*Figura 13.* Tapas laterales.

Las dimensiones de la caja que resguarda los componentes electrónicos y sus indicadores está basada en función de los módulos Arduino, GPS, SD y la batería, dando así como resultado las siguientes medidas: 130 mm de largo, 65 mm de ancho y 60 mm de alto como se indica en la figura 14. Los planos de construcción de la caja se muestran en el Anexo B.

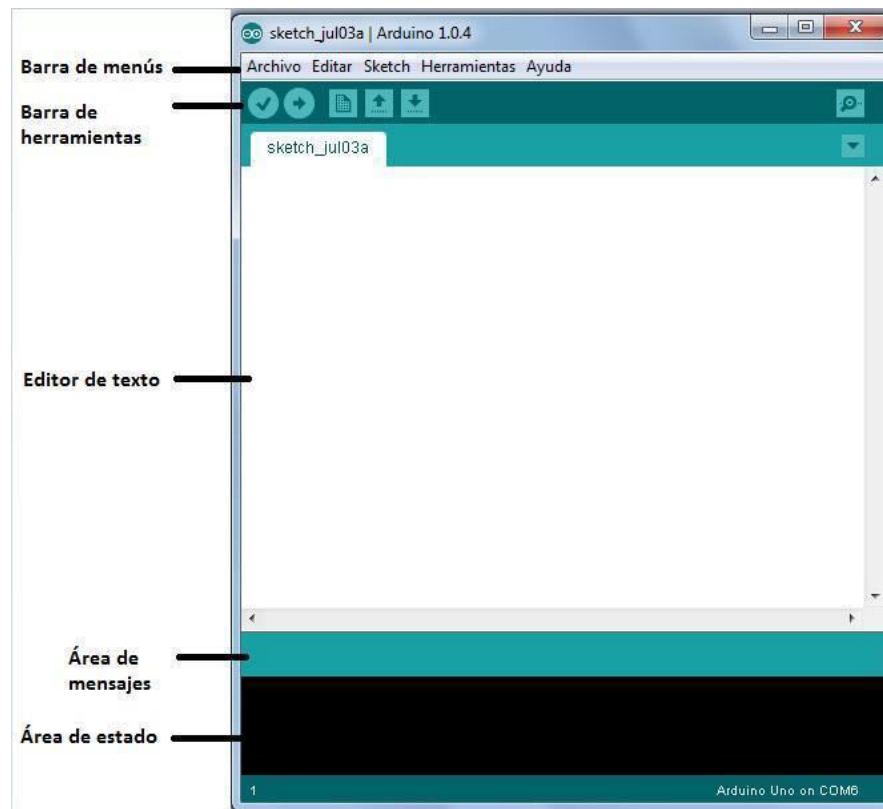


*Figura 14.* Dimensiones de la caja de acrílico.

### Diseño del software

La Interfaz de desarrollo para el diseño del software es la versión 1.0.4. El programa es de libre acceso y se lo puede descargar de la página web oficial de Arduino. (ARDUINO, 2006)

El software de Arduino muestra una interfaz constituida por un editor de textos donde se puede desarrollar la programación, una barra de herramientas y de menús que permiten un trabajo óptimo, área de estado y área de mensajes. (Figura 15)



*Figura 15.* Interfaz del software Arduino 1.0.4.

El software cuenta con librerías pre establecidas que ayudan al programador a elaborar los programas de una forma más eficaz. Las librerías que se utilizaron durante el desarrollo de este proyecto son: SD.h, GPSconfig.h y SoftwareSerial.h. El detalle de la programación se muestra en el Anexo C al final del presente trabajo.

El diagrama de flujo que resume todo el proceso que realiza la programación en el módulo Arduino se indica en la figura 16.

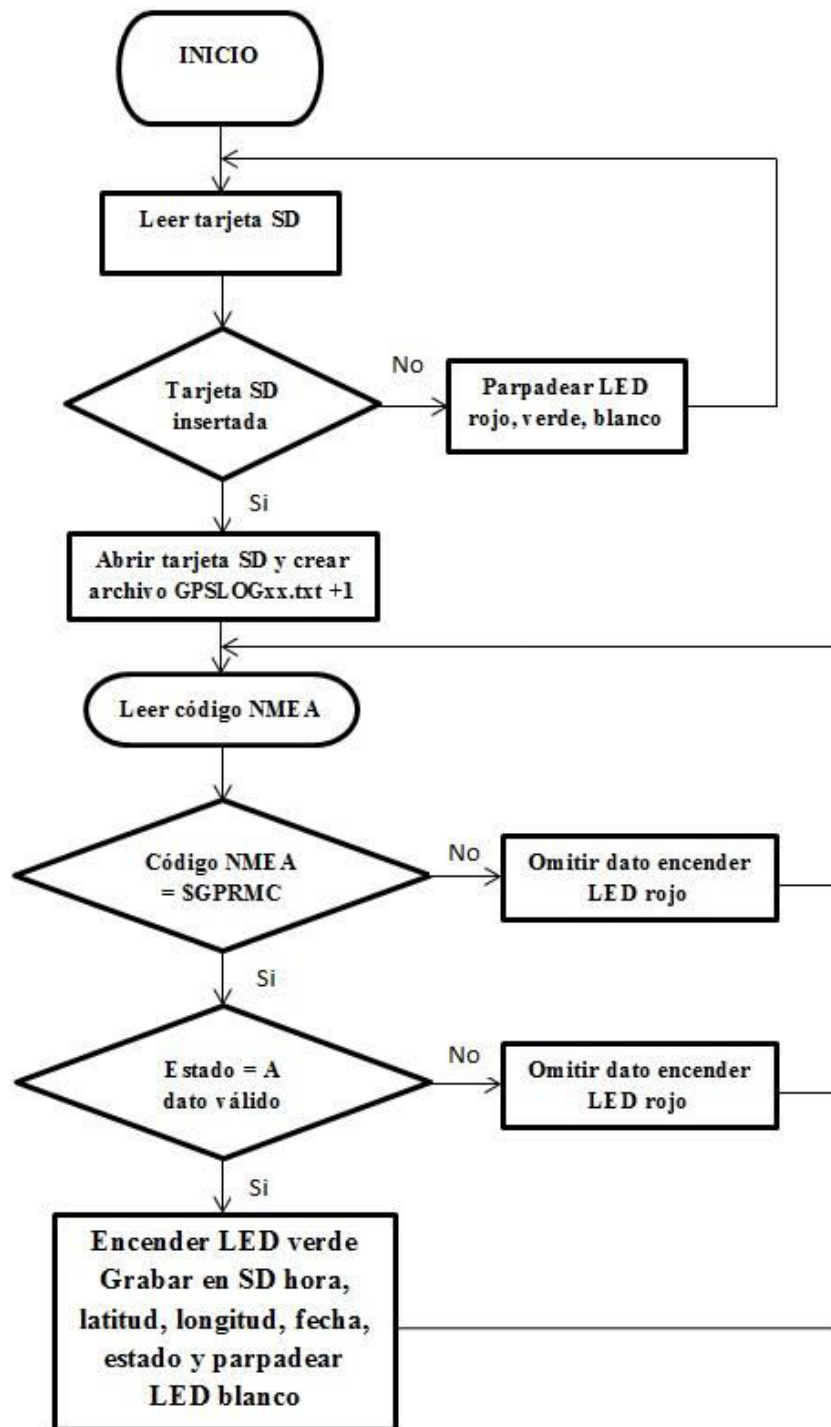


Figura 16. Diagrama de flujo del módulo de almacenamiento de coordenadas GPS.

## Procesamiento de datos en el programa GPS Visualizer y Google Earth

Al recolectar los datos en la tarjeta SD, estos deben ser procesados en el programa GPS Visualizer. Esta es una herramienta en línea que crea mapas y perfiles de datos geográficos. La entrada puede ser en forma de datos GPS (tracks y waypoints), rutas de conducción, direcciones de calles o coordenadas simples. (SCHNEIDER A. 2013)

Para crear un archivo KML en el programa GPS Visualizer, este se debe configurar para que los datos sean mostrados según se lo desee. La ventana principal de este programa se muestra en la figura 17.

Figura 17. Ventana principal GPS Visualizer.

El archivo .txt que se almacena en la tarjeta de memoria debe ser cargado mediante la opción “examinar” que se muestra en la parte superior de la ventana principal del programa GPS Visualizer (ver Figura 17).

El menú Waypoint Options permite configurar diferentes formas para los puntos recolectados permitiendo una mejor identificación en el mapa, esto se logra cambiando la opción en “Default Icon”; de igual manera se puede dar un matiz de colores utilizando la opción “Colorize Points Using”, como se muestra en la figura 18.

**Waypoint options** [hide advanced options \[-\]](#)

Waypoint labels:  ?

Default icon:  ? Icon color:  ?

Waypoint opacity:  ? Waypoint scale:  ?

Preserve waypoint colors and symbols (e.g., from KML or Garmin):  ?

Colorize points using:  ? Colorize labels:  ?

Synthesize name:  (field names in {curly} brackets) ?

Synthesize description:  (fields in {curly} brackets) ?

Synthesize folders:  (fields in {curly} brackets) ?

Show waypoints:  ?

Altitude mode:  ?

Discard outliers:  ?

Show only points whose 'name' matches this pattern:  ?

Garmin waypoint icons:  ?

Include driving directions links in info windows:  ?

Geotag points with times but no positions:  ?

Geotagging offset:  hours (positionless points vs. GPS data) ?

*Figura 18.* Menú de configuración GPS Visualizer.

Una vez finalizada la configuración del programa, se procede a crear el archivo KML dando clic sobre la opción “Create KML File” (ver Figura 17), el cual se debe guardar en el computador para poder visualizar los puntos en el programa Google Earth.

Google Earth es un software libre proporcionado por Google, en el cual se permite visualizar cualquier parte del mundo a través de un globo terráqueo virtual y ver imágenes de satélite, mapas, relieve y edificios 3D, entre otras cosas. (GOOGLE EARTH, 2013)

La ventana principal de Google Earth muestra un globo terráqueo sobre el cual se puede movilizar y buscar sitios sin necesidad de ingresar coordenadas, prácticamente se puede realizar un paseo virtual. (Figura 19)



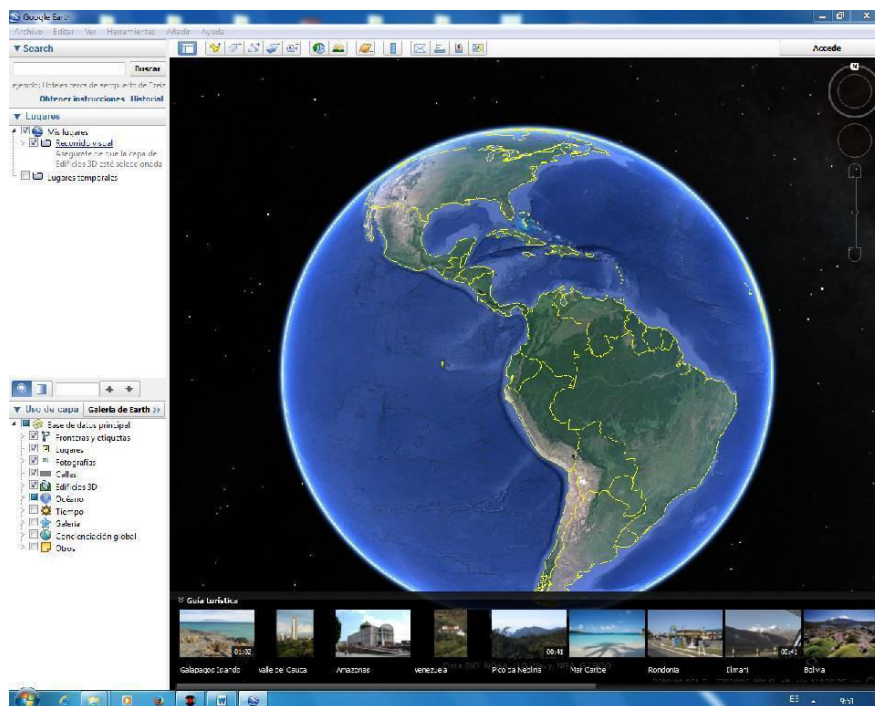


Figura 19. Ventana principal del programa Google Earth.

El archivo KML puede ser abierto directamente dando doble clic sobre el mismo, o a su vez abrirlo desde el programa Google Earth, el cual trazará automáticamente los puntos obtenidos por el módulo GPS como se indica en el ejemplo de la figura 20.

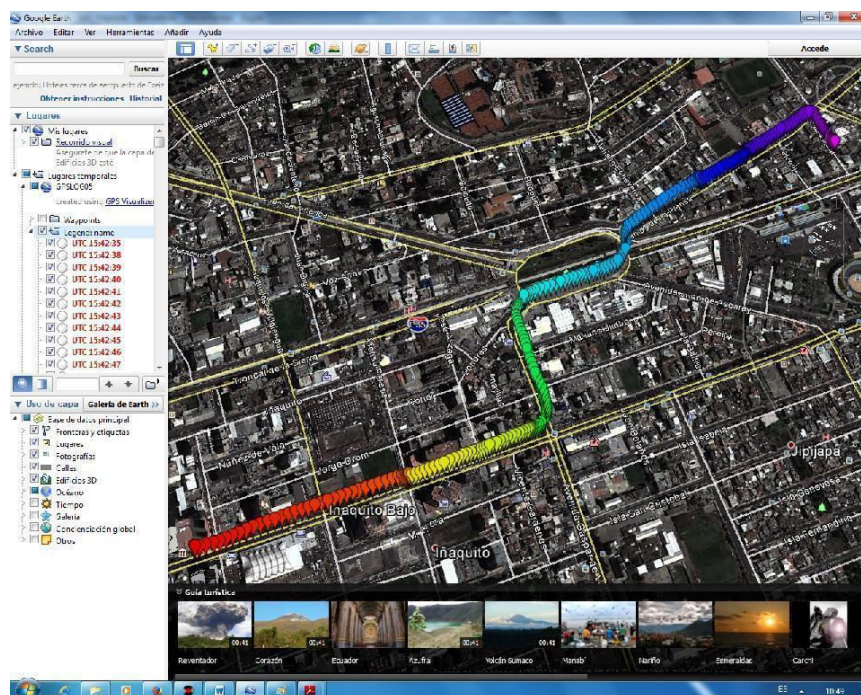


Figura 20. Ejemplo de una ruta trazada en Google Earth.

## **Experimentos**

Se realizaron 2 experimentos utilizando un GPS convencional de la marca Magellan 315 y el módulo construido para la verificación de datos obtenidos.

### **Experimento 1**

Consistió en ubicar los 2 GPS en un mismo punto y obtener coordenadas durante 2 minutos, los datos fueron recabados cada 15 segundos para su posterior análisis. Este experimento busca comprobar la fiabilidad de los datos y el margen de error respecto del punto de ubicación original.

### **Experimento 2**

Este experimento se basó en realizar un recorrido previamente establecido 3 veces y sobre el cual se obtuvieron datos del GPS Magellan 315 y el módulo GPS en puntos arbitrarios del recorrido para un posterior análisis. La ruta establecida fue realizada en un automóvil particular alrededor del parque la Carolina en la ciudad de Quito, el cual se encuentra rodeado por las siguientes avenidas: Naciones Unidas, Shyris, Eloy Alfaro, República, Amazonas y Japón. Como punto de partida y llegada se tomó la Av. Naciones Unidas y Shyris.

El GPS Magellan 315 tiene las siguientes características de trabajo (MANUAL OWL .2013):

- El receptor utiliza la tecnología de 12 canales paralelos, que permiten ubicar más de 12 satélites y actualizar la información con una antena cuadrifilar.
- El tiempo de adquisición de datos en condiciones óptimas de trabajo (cielo abierto), es de aproximadamente 15 segundos en “caliente” y de 1 minuto en “frío”, con actualización de datos cada segundo.
- El margen de error de la posición es de 15 metros aproximadamente.



## Resultados

### Experimento 1

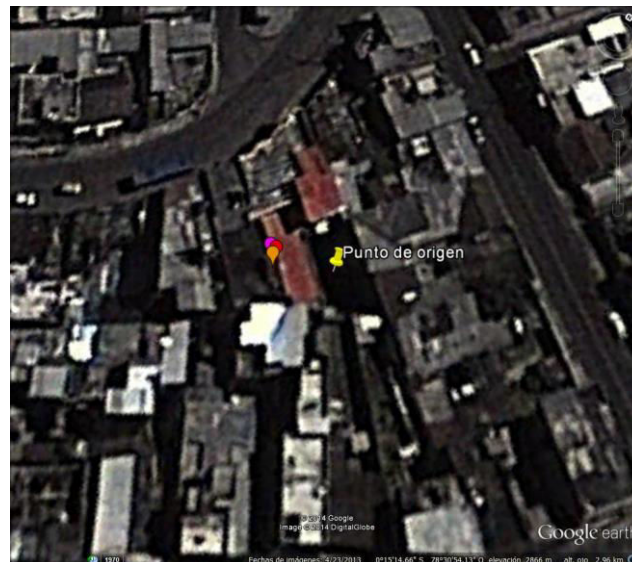
Los resultados de los datos obtenidos por el módulo GPS y el GPS Magellan 315 se muestran en la tabla 2, tomando en cuenta que cada 15 segundos representa una muestra.

Tabla 2

*Resultados de muestras GPS sobre el mismo punto*

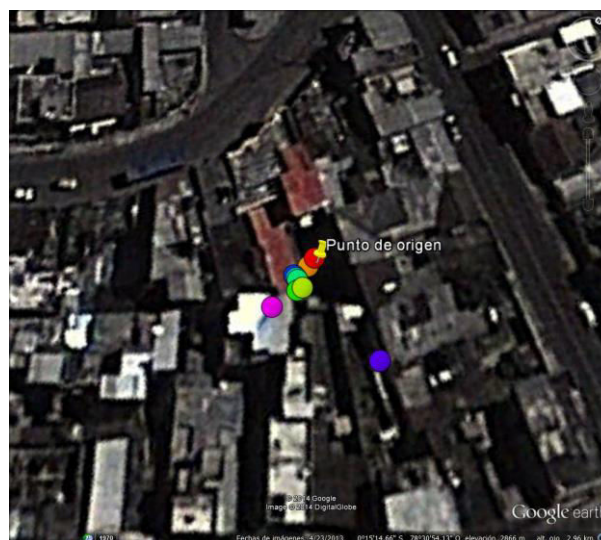
Dato	Hora	GPS Magellan 315		Módulo GPS Arduino	
		Latitud °	Longitud °	Latitud °	Longitud °
1	9:18:00	-00.25408	-78.51512	-00.25405	-78.51505
2	9:18:15	-00.25409	-78.51512	-00.25406	-78.51505
3	9:18:30	-00.25408	-78.51512	-00.25410	-78.51504
4	9:18:45	-00.25408	-78.51513	-00.25410	-78.51504
5	9:19:00	-00.25408	-78.51513	-00.25409	-78.51505
6	9:19:15	-00.25408	-78.51513	-00.25409	-78.51506
7	9:19:30	-00.25408	-78.51513	-00.25409	-78.51506
8	9:19:45	-00.25408	-78.51513	-00.25413	-78.51488
9	9:20:00	-00.25408	-78.51513	-00.25415	-78.51506

Los datos completos obtenidos por el GPS Magellan 315 se muestran en el Anexo D y su representación gráfica en la figura 21. Cada uno de los datos obtenidos muestran un color diferente y con una forma tipo paleta; en el que en dato 1 empieza con el color rojo y termina en el dato 9 con color fucsia.



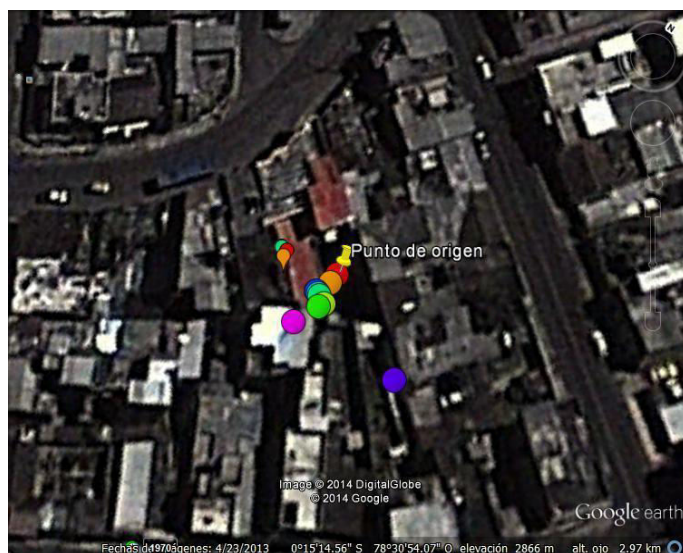
*Figura 21.* Datos GPS Magellan 315.

Los datos obtenidos por el Módulo se indican en la figura 22. Cada uno de los datos obtenidos muestran un color diferente y con una forma circular; en el que en dato 1 empieza con el color rojo y termina en el dato 9 con color fucsia.



*Figura 22.* Datos del Módulo GPS.

La combinación de los 2 datos y la marca de origen se muestra en la figura 23, tomando en cuenta que las formas tipo paleta representan los datos del GPS Magellan 315 y los círculos a los datos del Módulo GPS.



*Figura 23. Combinación de datos GPS y marca de origen.*

En la tabla 3 se muestra la distancia de cada uno de los datos respecto al punto de origen del cual se realizó esta prueba y su respectivo promedio (Margen de error).

Tabla 3

*Distancia entre el punto de origen y cada dato GPS.*

Dato	Distancia en metros desde el punto de origen	
	GPS Magellan 315	Módulo GPS
1	8.55	0.64
2	8.69	2.39
3	8.36	6.24
4	9.66	7.12
5	9.66	5.26
6	9.82	5.30
7	9.51	5.71
8	7.73	20.83
9	8.35	12.91
Promedio	8.92	7.37

## Experimento 2

### Recorrido 1.

Los datos obtenidos por el Módulo GPS y el GPS Magellan 315 se muestran en la tabla 4, el muestreo se realizó en puntos arbitrarios mientras se recorría la ruta, con punto de inicio y fin la Av. Naciones Unidas y Shyris. El dato 1 representa el inicio del recorrido y su hora de partida, mientras que el dato 11 representa la hora de llegada y el final del recorrido.

Tabla 4

*Resultados de muestras GPS del primer recorrido.*

Dato	Hora	GPS Magellan 315		Módulo GPS Arduino	
		Latitud °	Longitud °	Latitud °	Longitud °
1	9:45:47	-00.17663	-78.48210	-00.17868	-78.48047
2	9:46:19	-00.17665	-78.48208	-00.17880	-78.48075
3	9:47:29	-00.17967	-78.48164	-00.17980	-78.48167
4	9:47:43	-00.18115	-78.48187	-00.18115	-78.48187
5	9:48:50	-00.18583	-78.48272	-00.18595	-78.48281
6	9:50:39	-00.18826	-78.48308	-00.18824	-78.48310
7	9:53:40	-00.18994	-78.48445	-00.19000	-78.48446
8	9:55:55	-00.18513	-78.48713	-00.18476	-78.48713
9	9:57:16	-00.17865	-78.48582	-00.17854	-78.48581
10	9:58:17	-00.17648	-78.48384	-00.17642	-78.48378
11	9:59:07	-00.17667	-78.48209	-00.17660	-78.48214

*Nota. Las imágenes de los datos obtenidos por el GPS Magellan 315 se muestran en el Anexo E.*

La representación gráfica de los datos de muestra del GPS Magellan 315 se indica en la figura 24 (círculos), y los de Módulo GPS en la figura 25 (paletas). En la figura 26 se puede apreciar todos los datos obtenidos por el Módulo GPS en conjunto con los datos de muestra del GPS Magellan 315 (paletas y círculos). El punto de inicio y fin está marcado con la forma de un automóvil.



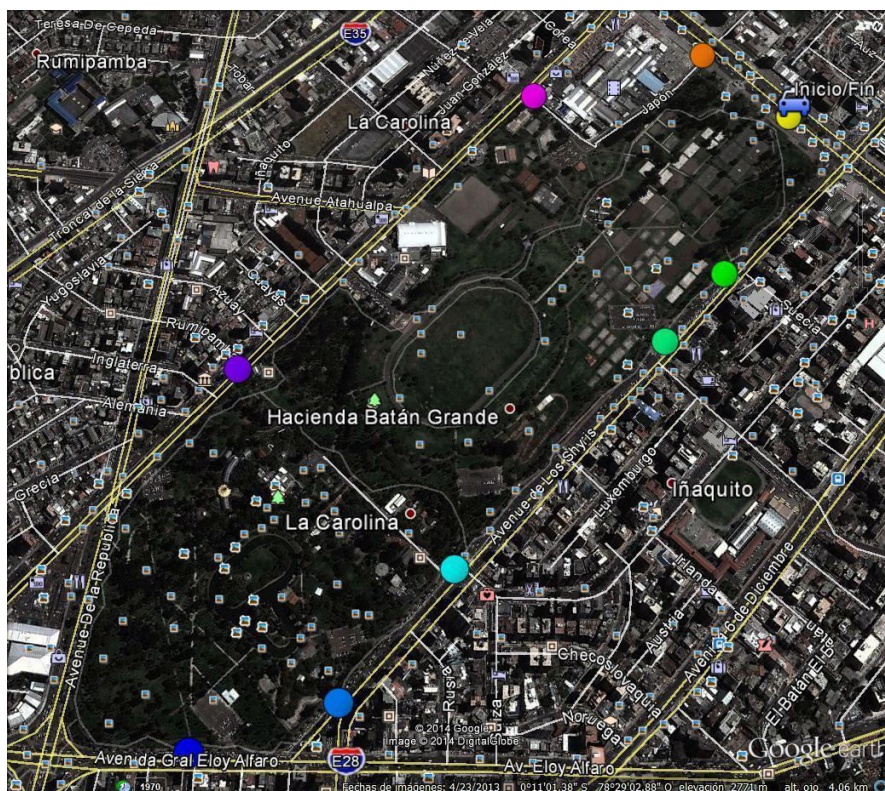


Figura 24. Gráfica de los datos de muestra del GPS Magellan 315.

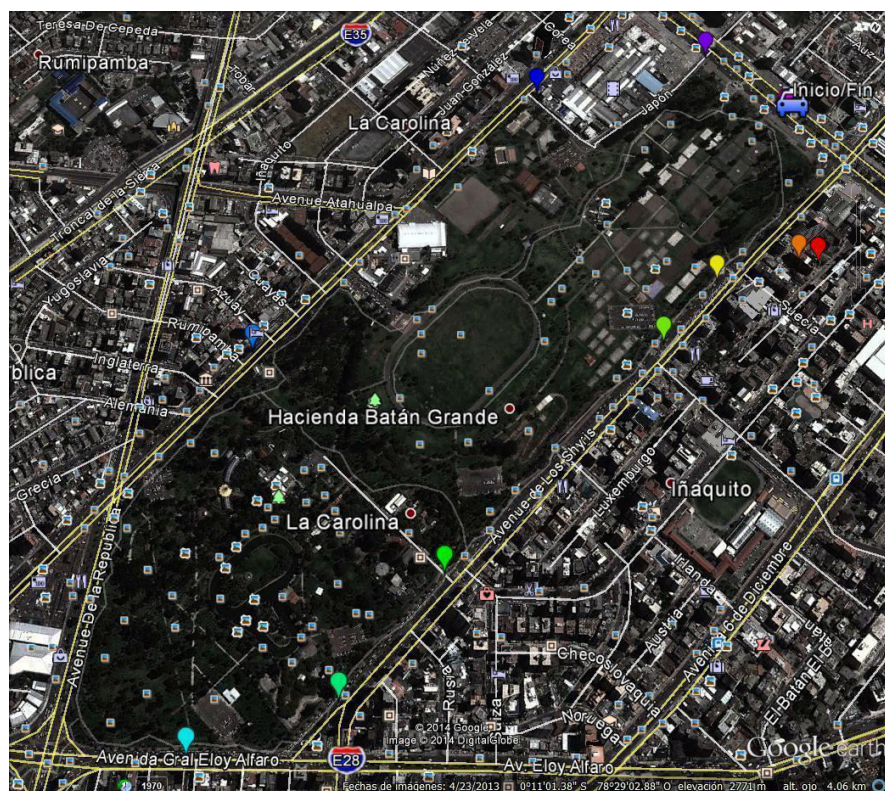
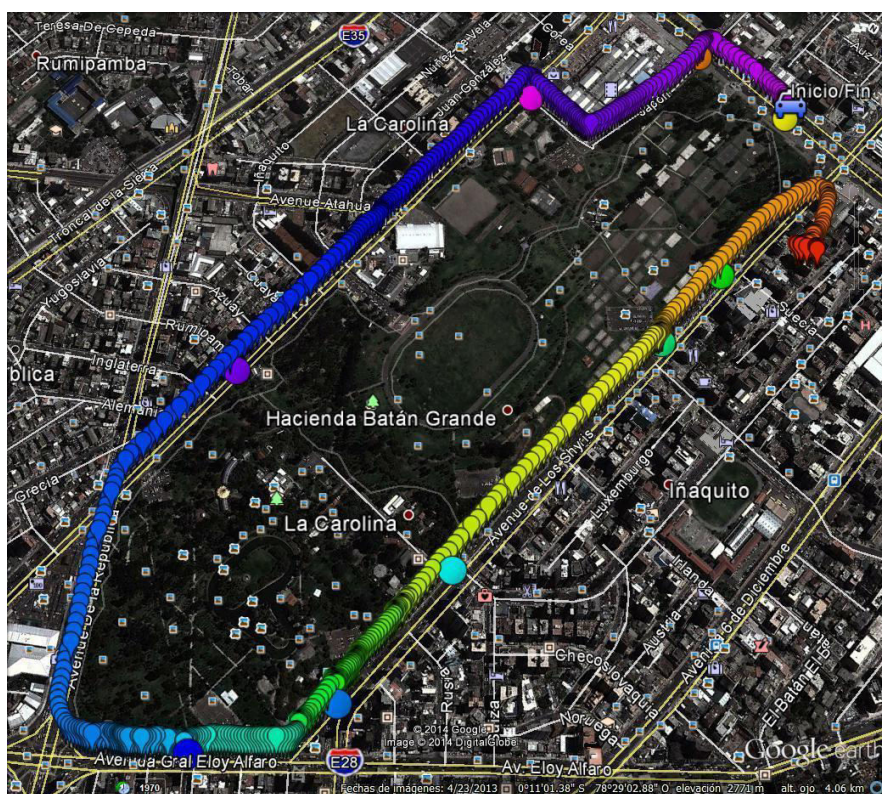


Figura 25. Gráfica de los datos de muestra del Módulo GPS.





*Figura 26.* Gráfica de los datos de muestra del GPS Magellan 315 y todos los datos del Módulo GPS

### **Recorrido 2.**

Los datos del segundo recorrido obtenidos por el Módulo GPS y el GPS Magellan 315 se muestran en la tabla 5, el muestreo se realizó en puntos arbitrarios mientras se recorría la ruta, con punto de inicio y fin la Av. Naciones Unidas y Shyris en el que el módulo se reseteo para iniciar el nuevo recorrido (forma de automóvil). Para este segundo recorrido, el muestreo se realizó en 14 datos que indican la hora de inicio y fin del mismo.

Tabla 5

*Resultados de muestras GPS del segundo recorrido.*

Dato	Hora	GPS Magellan 315		Módulo GPS Arduino	
		Latitud °	Longitud °	Latitud °	Longitud °
1	9:59:36	-00.17666	-78.48210	-00.17667	-78.48215
2	10:00:40	-00.18086	-78.48182	-00.18092	-78.48184
3	10:01:33	-00.18449	-78.48249	-00.18455	-78.48248
4	10:02:09	-00.18861	-78.48322	-00.18861	-78.48318
5	10:02:18	-00.18882	-78.48324	-00.18877	-78.48324
6	10:04:01	-00.18981	-78.48440	-00.18984	-78.48433
7	10:04:59	-00.19062	-78.48559	-00.19061	-78.48546
8	10:06:22	-00.18899	-78.48694	-00.18899	-78.48695
9	10:07:13	-00.18606	-78.48730	-00.18594	-78.48732
10	10:07:17	-00.18557	-78.48722	-00.18545	-78.48723
11	10:08:13	-00.17978	-78.48615	-00.17972	-78.48622
12	10:09:09	-00.17879	-78.48504	-00.17874	-78.48514
13	10:09:36	-00.17684	-78.48381	-00.17675	-78.48385
14	10:10:16	-00.17666	-78.48197	-00.17665	-78.48201

*Nota. Los datos obtenidos por el GPS Magellan 315 se muestran en el Anexo F.*

La representación gráfica de los datos de muestra del segundo recorrido se indica en la figura 27 (círculos) que corresponden al GPS Magellan 315, y los de Módulo GPS en la figura 28 (paletas). En la figura 29 se puede apreciar todos los datos obtenidos por el Módulo GPS en conjunto con los datos de muestra del GPS Magellan 315 (paletas y círculos).



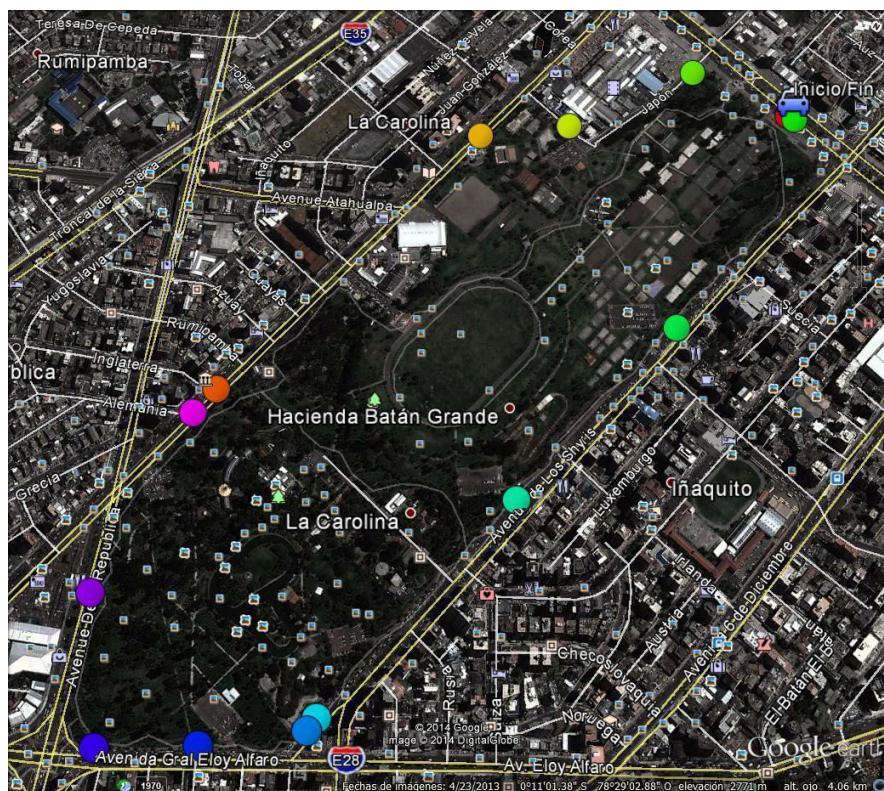


Figura 27. Gráfica de los datos de muestra del GPS Magellan 315.

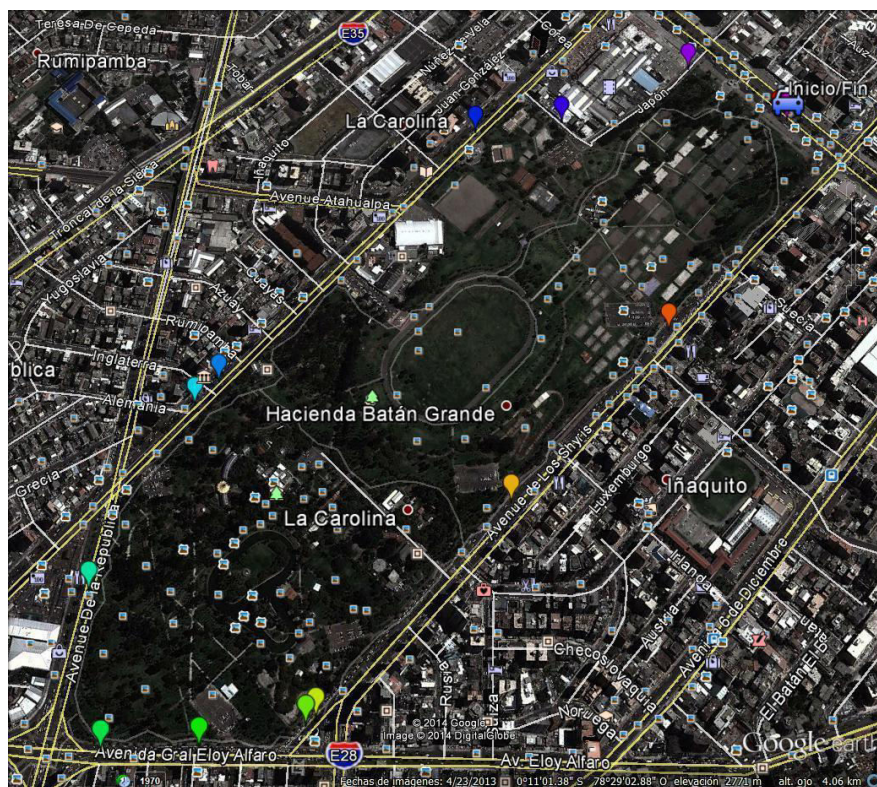
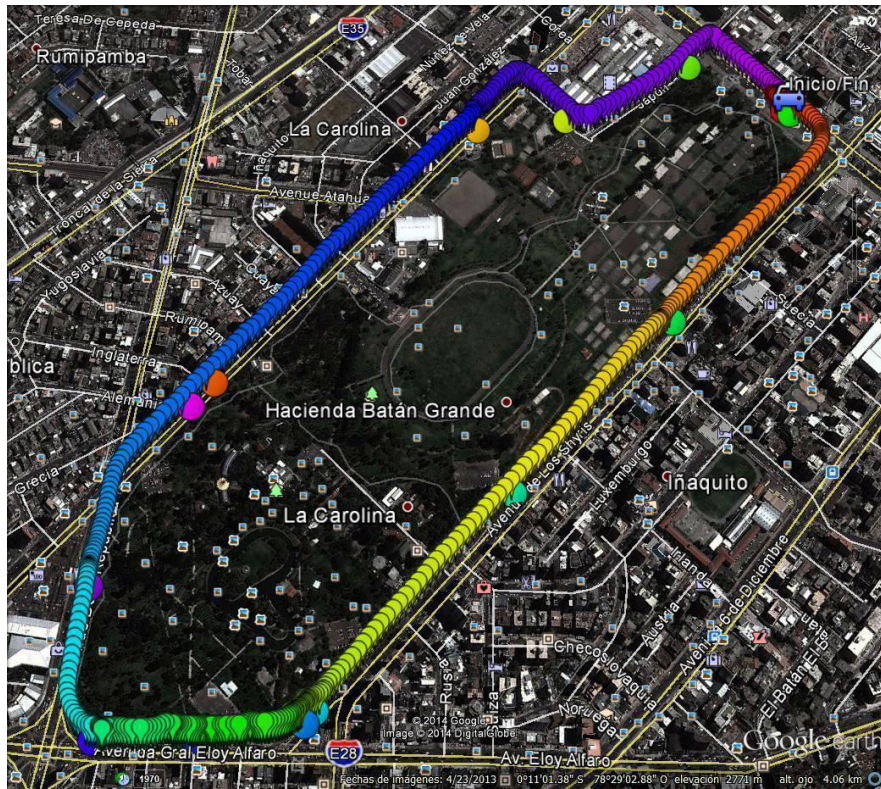


Figura 28. Gráfica de los datos de muestra del Módulo GPS.





*Figura 29.* Gráfica de los datos de muestra del GPS Magellan 315 y todos los datos del Módulo GPS

### **Recorrido 3.**

En el tercer y último recorrido los datos obtenidos por el Módulo GPS y el GPS Magellan 315 se muestran en la tabla 6, de igual manera que en los anteriores recorridos se tomó muestras en puntos arbitrarios mientras se recorría la ruta, con punto de inicio y fin la Av. Naciones Unidas y Shyris en el que el módulo se reseteo para iniciar el nuevo recorrido. Para este último recorrido, el muestreo se realizó en 16 datos que indican la hora de inicio y fin del mismo.

Tabla 6

*Resultados de muestras GPS del tercer recorrido.*

Dato	Hora	GPS Magellan 315		Módulo GPS Arduino	
		Latitud °	Longitud °	Latitud °	Longitud °
1	10:10:44	-00.17665	-78.48198	-00.17668	-78.48200
2	10:11:32	-00.17993	-78.48170	-00.18006	-78.48168
3	10:11:52	-00.18101	-78.48188	-00.18106	-78.48182
4	10:12:59	-00.18636	-78.48282	-00.18645	-78.48287
5	10:13:23	-00.18775	-78.48306	-00.18776	-78.48305
6	10:15:18	-00.18992	-78.48440	-00.18996	-78.48448
7	10:15:27	-00.19008	-78.48456	-00.19008	-78.48473
8	10:16:53	-00.18949	-78.48662	-00.18939	-78.48677
9	10:17:34	-00.18602	-78.48721	-00.18589	-78.48731
10	10:17:59	-00.18318	-78.48672	-00.18300	-78.48683
11	10:18:53	-00.17907	-78.48604	-00.17894	-78.48611
12	10:19:11	-00.17878	-78.48520	-00.17868	-78.48523
13	10:19:31	-00.17874	-78.48458	-00.17866	-78.48458
14	10:20:07	-00.17640	-78.48381	-00.17638	-78.48376
15	10:20:42	-00.17655	-78.48232	-00.17658	-78.48215
16	10:21:12	-00.17663	-78.48215	-00.17665	-78.48207

*Nota. Los datos obtenidos por el GPS Magellan 315 se muestran en el Anexo G.*

La representación gráfica de los datos de muestra del tercer recorrido se indica en la figura 30 (círculos) que corresponden al GPS Magellan 315, y los de Módulo GPS en la figura 31 (paletas). En la figura 32 se puede apreciar todos los datos obtenidos por el Módulo GPS en conjunto con los datos de muestra del GPS Magellan 315 (paletas y círculos).



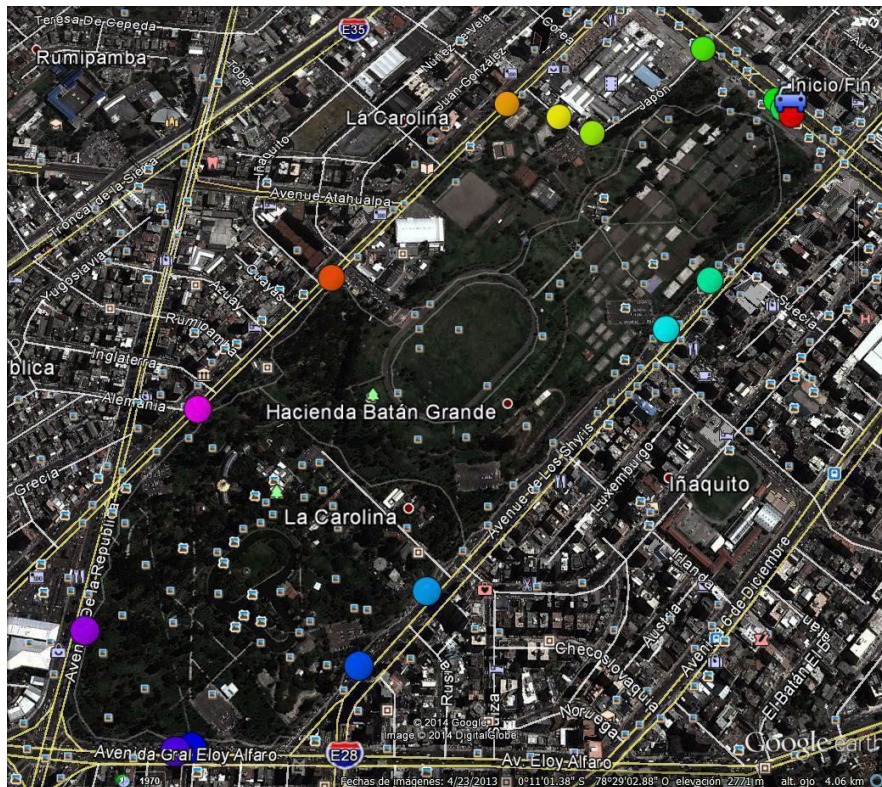


Figura 30. Gráfica de los datos de muestra del GPS Magellan 315.

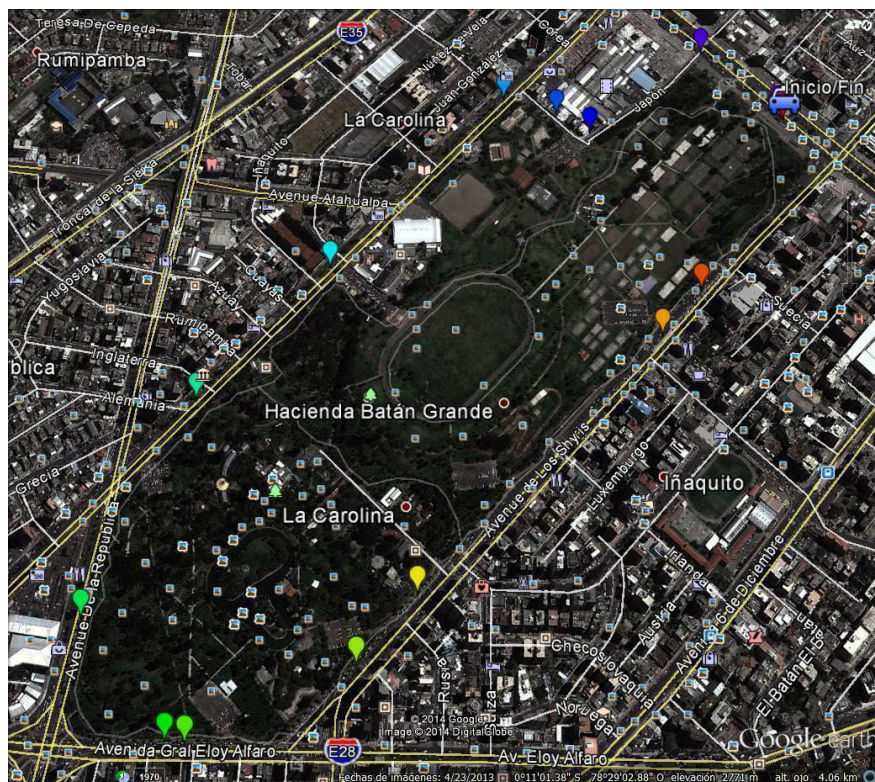
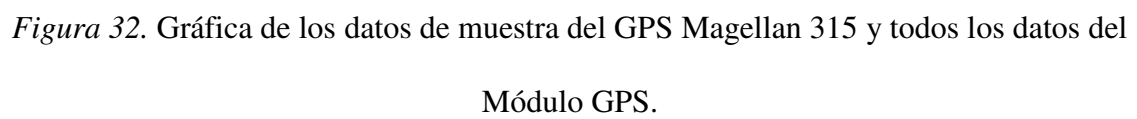


Figura 31. Gráfica de los datos de muestra del Módulo GPS.





## Discusión

### Experimento 1

A pesar de que los 2 GPS se ubican sobre el mismo punto, se pudo apreciar que cada uno está desplazado del punto de origen como se indicó en la figura 23. Los datos correspondientes al GPS Magellan 315 se muestran con poca variación respecto uno de otro, mientras que los datos Obtenidos con el Módulo GPS se muestran más dispersos entre sí.

Además, se pudo comprobar que los datos obtenidos por los 2 GPS muestran un error en cuanto a la ubicación, dependiendo básicamente del clima y la visibilidad del espacio aéreo ya que se necesita un mínimo de 4 satélites disponibles para que los GPS puedan realizar una triangulación exacta. Para el GPS proporcionado por Dexter Industries que se utilizó en el Módulo, el margen de error de proximidad es de 2.5 metros proporcionado de fábrica (DEXTER INDUSTRIES, 2013); mientras que para el GPS Magellan 315 está determinado en función de su programación y el cual se despliega en la pantalla según el punto de ubicación; este puede variar entre 0 y 40 metros.

En la tabla 3 se muestran los datos de distancia entre el punto de origen y las muestras, en el que el promedio de error del GPS Magellan 315 es de 8.92 metros mientras que el del Módulo GPS es de 7.37 metros. A pesar de que los datos del GPS Magellan 315 son menos dispersos, el Módulo GPS muestra un dato más confiable en cuanto al margen de error.

El margen de error promedio significa que los datos de precisión mostrados por los 2 GPS son aceptables, tomando en cuenta que son dispositivos que funcionan en base a triangulación de datos y que este margen de error se puede dar en función de la cantidad de decimales tomados en los datos de latitud y longitud. Estos datos pueden variar en función de condiciones climáticas, recepción de señal reflejada o una triangulación con el mínimo de satélites disponibles.

A pesar de que el GPS del Módulo no posee una antena externa, los datos obtenidos fueron verificables y totalmente factibles de usar ya que su margen de error es aproximado a los datos que da el fabricante.

## **Experimento 2**

### **Recorrido 1.**

En la tabla 4 se pudo observar que existen datos coincidentes tanto en latitud como en longitud. El dato coincidente en latitud es el número 4, mientras que en longitud los datos coincidentes son 4 y 8. El dato número 4 es una muestra en particular en el que los datos coinciden tanto del GPS Magellan 315 como el Módulo GPS en el que se puede determinar que los 2 GPS pueden dar mediciones iguales.

En este recorrido se puede apreciar los datos de muestras del GPS Magellan 315 que se encuentran conformes a la ruta establecida tomando en cuenta el margen de error del mismo. Los puntos establecidos no se disipan o se muestran fuera de un rango de tolerancia normal como se pudo observar en la figura 24.

En la figura 25, los datos del Módulo GPS en su mayor parte se indican sobre la ruta establecida. Los 2 primeros datos de muestra se encuentran fuera de la ruta establecida, concluyendo que el Módulo GPS después de haberse encendido por primera vez debe esperar un lapso de tiempo pequeño hasta determinar el punto exacto de ubicación. Esto se puede determinar observando la figura 25 en la que los datos posteriores al primer y segundo punto de muestra se encuentran sobre la ruta trazada originalmente.

En la figura 26 se muestran todos los puntos obtenidos por el Módulo GPS y el GPS Magellan 315, en el cual se comprobó que el Módulo GPS necesita esperar un lapso de tiempo aproximado de 1 minuto para determinar el punto exacto después de haber sido encendido por primera vez.

De esta manera se puede corroborar los datos obtenidos en la prueba 1, en la que el módulo necesita un tiempo después de encenderse por primera vez para obtener datos con mayor fidelidad y precisión. El margen de error presentado por este motivo es de 290 metros aproximadamente respecto al punto marcado de inicio y fin.

En el primer recorrido el GPS Magellan 315 se comportó de una manera más precisa en comparación con el Módulo GPS, ya que no requirió de un tiempo para dar su ubicación y debido a que es un instrumento de uso profesional.

El Módulo GPS después del lapso de tiempo estipulado para trabajar en caliente, se comporta de una forma más precisa y conforme a los datos comparados con el GPS Magellan 315. De esta manera se concluye que el Módulo GPS debe trabajar en caliente para evitar lecturas erróneas a pesar de que los datos obtenidos se empiecen a guardar en la tarjeta SD.

### **Recorrido 2.**

Este recorrido empezó con el reseteo del módulo y el último punto receptado anteriormente, para ello el Módulo GPS ya estuvo trabajando en caliente. Los datos obtenidos en la tabla 5 muestran coincidencias en latitud y longitud. Los datos coincidentes en latitud fueron 4 y 8, mientras que en longitud se obtuvo el dato 5. La mayor cantidad de datos del Módulo GPS varía en pocos grados ya sea en forma ascendente o descendente respecto al GPS Magellan 315, en algunos casos muestra la diferencia de una diezmilésima. Estos datos revelan que cuando el Módulo GPS cuando trabaja “en caliente”, es más preciso.

En la figura 27 se puede apreciar que el trazo realizado por los datos del GPS Magellan 315 fue sobre la ruta establecida. La gráfica que marca el inicio y fin del recorrido se superpone sobre 2 datos que corresponde al primer y último dato de muestra tomada en la que el margen de error es mínimo, comprobando la precisión que tiene el instrumento.

Los datos correspondientes al Módulo GPS se muestran en la figura 28, en la que se puede apreciar que se afinó la precisión del Módulo GPS, debido a que este ya no trabaja “en

frío”. Los puntos de la gráfica ya no muestran un desfase en su inicio como en el recorrido 1, para este caso la precisión de la lectura es más efectiva ya que la marca de inicio y fin se superpone sobre el primer y último dato obtenido.

La representación gráfica de la figura 29 muestra los datos obtenidos en conjunto, en el que el trazo de la ruta se ve mucho más precisa con todos los puntos captados por el Módulo GPS y sin los desfases que se obtuvo en la ruta 1. La marca de inicio y fin se superpone en el primer y último dato obtenido por ambos GPS, dando como resultado una muestra más real en cuanto al recorrido planteado inicialmente

En la misma figura 29 también se puede apreciar que en cada una de las curvas se aprecia un trazo redondeado, esto es debido al margen de error presente en cada uno de los instrumentos, pero que para efectos prácticos de la ruta trazada no es de mayor influencia sobre el resultado general que es trazar la ruta previamente establecida.

### **Recorrido 3.**

Este último recorrido, al igual que los 2 recorridos anteriores, muestra datos coincidentes entre latitud y longitud, siendo los datos 7 y 13 respectivamente para cada caso. Adicional se muestra una diferencia entre dato y dato de diezmilésimas, siendo así de mayor precisión que los 2 anteriores recorridos (ver tabla 6).

La figura 30 y 31 despliegan imágenes de los datos obtenidos como muestras, en el que la diferencia es mínima entre el recorrido planteado y la ubicación de cada punto georeferenciado, corroborando así la precisión de los instrumentos.

La ruta grabada en la memoria SD que corresponde al tercer recorrido, se muestra en la figura 32 junto con los datos de muestra del GPS Magellan 315. Esta es una muestra práctica de que los datos obtenidos en el Módulo GPS se apegan a la ruta establecida y con una mínima variación con el otro GPS.



## **Conclusiones**

Estas pruebas determinaron que el Módulo GPS cumple con el objetivo de recibir los datos y almacenarlos en una memoria SD que permite su posterior análisis.

El Módulo GPS también cumple con los 2 principales pasos que se dan en el GIS, el primero que es recolectar datos y el segundo que es procesarlos en un computador. Estos 2 pasos como se indica en la figura 1, son los que a futuro sirven para prestación de servicios y mediante el cual se puede obtener ganancias monetarias.

De una forma práctica y visual se demostró la versatilidad con el que el Módulo GPS puede ser transportado para la obtención de datos ya sea a pie o usando un medio de transporte convencional.

## **Recomendaciones**

En la programación, se recomienda utilizar los pines adecuados para que cada uno de los módulos funcione correctamente y que no existan conflictos al momento de ejecución del programa.

La programación debe estar estructurada secuencialmente y para ello se recomienda realizar un diagrama de flujo para una mejor perspectiva al momento de programar y utilizar los datos obtenidos.

Durante todo el proceso de programación y pruebas, la batería que viene incluida en el Módulo GPS de Dexter Industries no debe ser retirada, esto causa que se pierda los datos de la constelación de satélites que vienen pre programadas. Si la batería es retirada, el módulo puede tardarse en recuperar estos datos alrededor de 3 días, siempre y cuando se lo deje conectado las 24 horas.

Se recomienda que antes de empezar a obtener datos reales en el Módulo GPS este permanezca encendido por lo menos un minuto antes y con recepción de datos correctos para que la ubicación sea más precisa al momento de obtener los datos de una ruta.

La duración de la batería en modo autónomo es de aproximadamente una hora y media, para lo cual es recomendable llevar consigo una de repuesto en caso de que los recorridos sean superiores a este tiempo.

El Módulo debe estar en lugares visibles, ya que carece de una antena externa, este puede perder datos o tomarlos erróneamente si se encuentra en un lugar cubierto.

### Referencias Bibliográficas

ANGULO ASATEGUI, J. & ANGULO MARTÍNEZ, I. (2003). *Micro controladores PIC: Diseño práctico de aplicaciones* (3ª edición). Madrid – España: McGRAW-HILL.

ARDUINO (2006). *Arduino Home*. Recuperado el 2 de septiembre del 2013 de: <http://www.arduino.cc/es/>

ARDUINO FORUM (2011). *Arduino GPS Shield*. Recuperado el 4 de octubre del 2013 de: <http://forum.arduino.cc/index.php?topic=51836.0>

BANZI, M. (2009). *Getting Started wiht Arduino* (First Edition). Sebastopol – Canada: O'Reilly.

BENCHIMOL, D (2011), *Microcontroladores*. Manuales Users. 213(1). 52–59.

BRICO GEEK (2011). *Tutorial Arduino: GPS Logger con EM406A, GPS Shield y MicroSD Shield*. Recuperado el 10 de septiembre del 2013 de: <https://www.sparkfun.com/datasheets/GPS/NMEA%20Reference%20Manual1.pdf>

BRICO GEEK (2011). *Tutorial Arduino: GPS Logger con EM406A, GPS Shield y MicroSD Shield*. Recuperado el 10 de septiembre del 2013 de: <http://blog.bricogeek.com/noticias/tutoriales/tutorial-arduino-gps-logger-con-em406a-gps-shield-y-microsd-shield/>

DEXTER INDUSTRIES (2013). *The GPS Shield for Arduino*. Recuperado el 2 de septiembre del 2013 de: [http://dexterindustries.com/Arduino-GPS\\_Shield.html](http://dexterindustries.com/Arduino-GPS_Shield.html)

DEXTER INDUSTRIES (2013). *User manual Wiki: GPS Shield for Arduino*. Recuperado el 2 de septiembre del 2013 de: <http://www.dexterindustries.com/manual/arduino-shields/gps-shield/>

EL-RABBANY, A (2002). *Introduction to GPS: The Global Positioning System* (First Edition). Norwood - MA: Artech House, Inc.

EXTREMADURA-WEB (2013). *Cableando SD LC Studio en Arduino Uno y Mega*. Recuperado el 18 de septiembre del 2013 de: <http://www.extremadura-web.es/Blog/2012/11/16/cableando-sd-lc-studio-en-arduino-uno-y-mega/>

GOOGLE EARTH (2013). *Google Earth*. Recuperado el 18 de septiembre del 2013 de: [http://www.google.es/intl/es\\_es/earth/](http://www.google.es/intl/es_es/earth/)

IBRAHIM, D. (2010). *SD Card Projects using the PIC Microcontroller* (First Edition). Burlington – USA: Newnes.

JUNE JAMRICH PARSONS. D. (2008). *Conceptos de Computación: Nuevas perspectivas* (10ª edición). Santa Fe – Colombia: Cengage Learning Editores S.A.

LATANDA ZARZOSA, N., & NÚÑEZ ANDRÉS, A. (2002). *Sistemas de Información Geográfica* (1ª edición). Barcelona - España: Ediciones UPC.

MANUAL OWL (2013). *Magellan GPS 315 User Manual*. Recuperado el 26 de septiembre del 2013 de: <http://www.manualowl.com/m/Magellan/GPS-315/Manual/240998>

MARGOLIS, M. (2011). *Arduino Cook Book* (First Edition). Sebastopol – Canada: O'Reilly.

PROFMASON (2012). *SD Card for the Arduino*. Recuperado el 26 de septiembre del 2013 de: <http://profmason.com/?p=1956>

SCHNEIDER A. (2013). *GPS Visualizer*. Recuperado el 26 de septiembre del 2013 de: <http://www.gpsvisualizer.com/>

SENPLADES (2013). *Objetivos del Plan Nacional del Buen Vivir*. Recuperado el 26 de septiembre del 2013 de: <http://plan.senplades.gob.ec/>

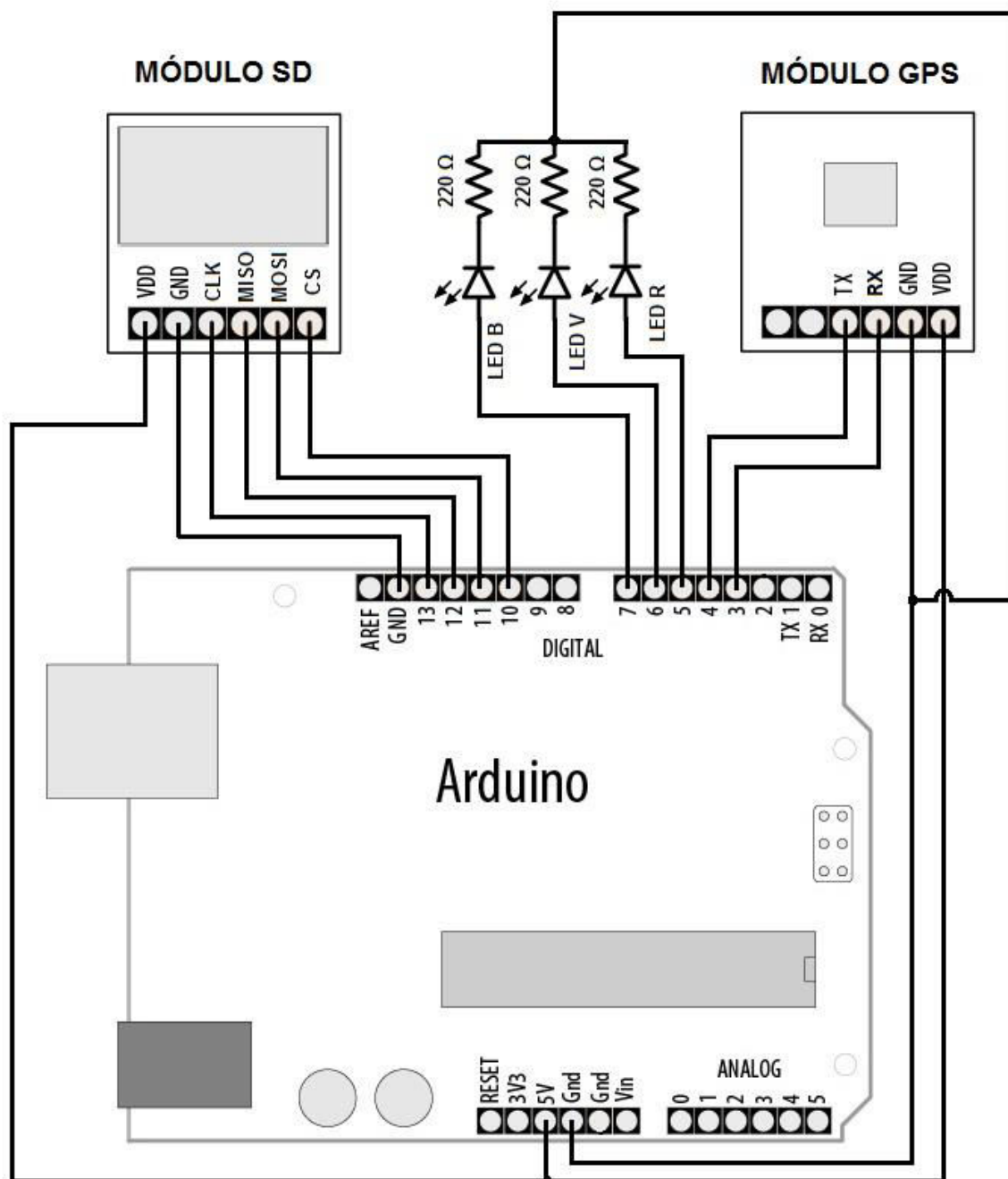
SIRF (2013). *NMEA Reference Manual*. Recuperado el 26 de septiembre del 2013 de:  
<http://www.sirf.com/>

TOMLINSON, R. (2007). *Thinking about GIS* (Third Edition). New York – USA: Ingram  
Publisher Services.

VALDÉS PÉREZ, F. & PALLÀS ARENY, R. (2007). *Microcontroladores: Fundamentos y  
Aplicaciones* (1ª edición). Barcelona – España: MARCOMBO S.A.

## Anexo A

## Diagrama de conexión eléctrica del Módulo GPS.







**Anexo C****Programación del Módulo GPS ejecutable en la versión 1.0.4 de Arduino**

```
////////////////////////////////////
```

```
//      Universidad Internacional SEK      //
```

```
//      Ingeniería Mecánica en Energía y Control      //
```

```
//Nombre: Módulo de Almacenamiento de coordenadas GPS //
```

```
//Autor: Israel Pazmiño                    //
```

```
//email: profesor_epn@yahoo.es           //
```

```
////////////////////////////////////
```

```
//Nombre de las Librerías
```

```
#include <SD.h>
```

```
#include <avr/sleep.h>
```

```
#include "GPSconfig.h"
```

```
#include <SoftwareSerial.h>
```

```
// Modo de ahorro de energía
```

```
#define LOG_RMC_FIXONLY 0
```

```
// Que es lo que se debe registrar
```

```
#define LOG_RMC 1 // RMC-Recommended Minimum Specific GNSS Data, message  
103,04
```

```
#define LOG_GGA 0 // GGA-Global Positioning System Fixed Data, message 103,00
```

```
#define LOG_GLL 0 // GLL-Geographic Position-Latitude/Longitude, message 103,01
```

```
#define LOG_GSA 0 // GSA-GNSS DOP and Active Satellites, message 103,02
```

```
#define LOG_GSV 0 // GSV-GNSS Satellites in View, message 103,03
```

```
#define LOG_VTG 0 // VTG-Course Over Ground and Ground Speed, message 103,05
```

```
// Pin digital Rx=3 Tx=4
```

```
SoftwareSerial gpsSerial = SoftwareSerial(3, 4);
```

```
// Ajuste a la velocidad de transmisión del módulo GPS.
```

```
#define GPSRATE 9600
```

```
// Ajuste de pines y definición
```

```
#define led1Pin 5
```

```
#define led2Pin 6
```

```
#define led3Pin 7
```

```
#define chipSelect 10
```

```
const int bufsize = 90;
```

```
char buffer[bufsize];
```

```
uint8_t bufferidx = 0;
```

```
bool fix = false; // Dato actual fijado
```

```
bool gotGPRMC; //Verdadero si el dato actual es una sentencia GPRMC
```

```
uint8_t i;
```

```
File logfile;
```

```
// Lee un valor hexadecimal y lo convierte en decimal
```

```
uint8_t parseHex(char c) {
```

```
    if (c < '0')
```

```
        return 0;
```

```
    if (c <= '9')
```

```
        return c - '0';
```

```
    if (c < 'A')
```

```
        return 0;
```

```
    if (c <= 'F')
        return (c - 'A')+10;
}

// Código para parpadear los LED

//si no hay una tarjeta de memoria insertada crear un mensaje de error
void error(uint8_t errno) {
    while(1) {
        for (i=0; i<errno; i++) {
            digitalWrite(led1Pin, HIGH);
            digitalWrite(led2Pin, HIGH);
            digitalWrite(led3Pin, HIGH);
            delay(100);
            digitalWrite(led1Pin, LOW);
            digitalWrite(led2Pin, LOW);
            digitalWrite(led3Pin, LOW);
            delay(100);
        }
        for (; i<10; i++) {
            delay(200);
        }
    }
}

//declaracion de variables

void setup() {
    WDTCSR |= (1 << WDCE) | (1 << WDE);
```

```
WDTCsr = 0;

Serial.begin(9600);

Serial.println("\r\nGPSlogger");

pinMode(led1Pin, OUTPUT);

pinMode(led2Pin, OUTPUT);

pinMode(led3Pin, OUTPUT);


// Definición del pin para la lectura de memoria SD

pinMode(10, OUTPUT);


// Revisa si la tarjeta SD esta insertada y si puede ser abierta
if (!SD.begin(chipSelect)) {

    Serial.println("Card init. failed!");

    error(1);

}

strcpy(buffer, "GPSLOG00.TXT"); //Empieza el archivo en 00

for (i = 0; i < 100; i++) {

    buffer[6] = '0' + i/10;

    buffer[7] = '0' + i%10;

    // Crear si no existe el archivo, no abrir el existente

    // Escribir y sincronizar después de abrir el archivo

    if (!SD.exists(buffer)) {

        break;

    }

}

logfile = SD.open(buffer, FILE_WRITE);

if( ! logfile ) {
```

```
    Serial.print("Couldnt create ");
    Serial.println(buffer);
    error(3);
}

Serial.print("Writing to ");
Serial.println(buffer);

// Conectar al GPS a la velocidad seleccionada
gpsSerial.begin(GPSRATE);
Serial.println("Ready!");
gpsSerial.print(SERIAL_SET);
delay(250);
}

void loop() {
    char c;
    uint8_t sum;

    // Lee una linea de datos
    if (gpsSerial.available()) {
        c = gpsSerial.read();
        if (bufferidx == 0) {
            while (c != '$')
                c = gpsSerial.read(); // espera a que empiece con el símbolo $
        }
        buffer[bufferidx] = c;
```

```
if (c == '\n') {  
    buffer[bufferidx+1] = 0;  
    if (buffer[bufferidx-4] != '*') {  
        // Si no existe checksum imprimir *  
        Serial.print('*');  
        bufferidx = 0;  
        return;  
    }  
  
    // Si existe checksum  
    sum = parseHex(buffer[bufferidx-3]) * 16;  
    sum += parseHex(buffer[bufferidx-2]);  
  
    // verificar checksum  
    for (i=1; i < (bufferidx-4); i++) {  
        sum ^= buffer[i];  
    }  
    if (sum != 0) {  
        Serial.print('~');  
        bufferidx = 0;  
        return;  
    }  
  
    // Dato correcto  
    gotGPRMC = strstr(buffer, "GPRMC");  
    if (gotGPRMC) {
```



```
// Comprobar si existe un dato GPRMC

}

if (LOG_RMC_FIXONLY) {

    if (!fix) {

        Serial.print('_');

        bufferidx = 0;

        return;

    }

}

Serial.print('#');

if (gotGPRMC) //Si se obtuvo un Dato GPRMC

{

    char *p = buffer; //dato a ser almacenado en el buffer

    p = strchr(p, ',')+1;

    buffer[0] = 'U';

    buffer[1] = 'T';

    buffer[2] = 'C';

    buffer[3] = ' ';

    buffer[4] = p[0];

    buffer[5] = p[1];

    buffer[6] = ':';

    buffer[7] = p[2];

    buffer[8] = p[3];

    buffer[9] = ':';

    buffer[10] = p[4];

    buffer[11] = p[5];

    // ignoramos milisegundos
```

```
buffer[12] = ',';

//Lee la primera coma de la sentencia NMEA

p = strchr(buffer+12, ',')+1;

p = strchr(p, ',')+1;

buffer[13] = p[0];

// Comprueba si es un dato correcto = A

//Enciende el LED 1 si es = V y apaga el LED 2

if (p[0] == 'V') {

    digitalWrite(led2Pin, LOW);

    digitalWrite(led1Pin, HIGH);

    fix = false;

}

//Enciende el LED 2 si es = A y apaga el LED 1

else {

    digitalWrite(led2Pin, HIGH);

    digitalWrite(led1Pin, LOW);

    fix = true;

    buffer[14] = ',';

    // Encuentra latitud

    //Y almacena datos en el buffer

    p = strchr(p, ',')+1;

    buffer[15] = '+';

    buffer[16] = p[0];

    buffer[17] = p[1];

    buffer[18] = ' ';

    strncpy(buffer+19, p+2, 7);

    buffer[26] = ',';
```

```
//comprueba si está en el sur o norte

//Si está en el sur coloca un signo menos

p = strchr(buffer+27, ',')+1;

if (p[0] == 'S')

    buffer[15] = '-';


// Encuentra longitud

//Y almacena datos en el buffer

p = strchr(p, ',')+1;

buffer[27] = '+';

buffer[28] = p[0];

buffer[29] = p[1];

buffer[30] = p[2];

buffer[31] = ' ';

strncpy(buffer+32, p+3, 7);

buffer[39] = ',';


//comprueba si está en el este u oeste

//Si esta en oeste coloca un signo menos

p = strchr(buffer+40, ',')+1;

if (p[0] == 'W')

    buffer[27] = '-';


//Omitir los siguientes datos que se encuentran entre comas

p = strchr(p, ',')+1;

p = strchr(p, ',')+1;

p = strchr(p, ',')+1;


//Leer dato de fecha y almacenar en buffer

buffer[40] = 'F';
```

```
buffer[41] = 'e';
buffer[42] = 'c';
buffer[43] = 'h';
buffer[44] = 'a';
buffer[45] = ':';
buffer[46] = ' ';
buffer[47] = p[0];
buffer[48] = p[1];
buffer[49] = '-';
buffer[50] = p[2];
buffer[51] = p[3];
buffer[52] = '-';
buffer[53] = p[4];
buffer[54] = p[5];
buffer[55] = ',';
bufferidx++;

digitalWrite(led3Pin, HIGH); // Enciende el LED 3 que indica que el dato se está
guardado

logfile.println("Name,Sym,Latitude,Longitude,Desc"); //escribe los nombres de los
datos almacenados en el buffer

logfile.write((uint8_t *) buffer, 55); //Escribe los datos almacenados en el buffer en la
tarjeta SD

logfile.println(" ");

Serial.println(buffer); //Muestra lo almacenado en el buffer en el monitor serial

logfile.flush(); //Vacía el búfer de entrada de datos en serie

digitalWrite(led3Pin, LOW); //Apaga el LED 3 que indica que el dato se ha guardado

bufferidx = 0; //Reestablece el punto de buffer
```

```
    }

    return;

} //if (gotGPRMC)

}

    // En caso de que el dato sea incorrecto despliega un signo de admiración en el
monitor serial

    bufferidx++;

    if (bufferidx == bufsize-1) {

        Serial.print('!');

        bufferidx = 0;

    }

}

else {

}

}
```

## Anexo D

## Datos del GPS Magellan 315 Experimento 1



Dato 1



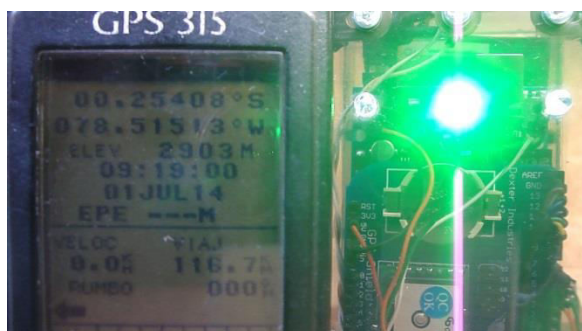
Dato 2



Dato 3



Dato 4



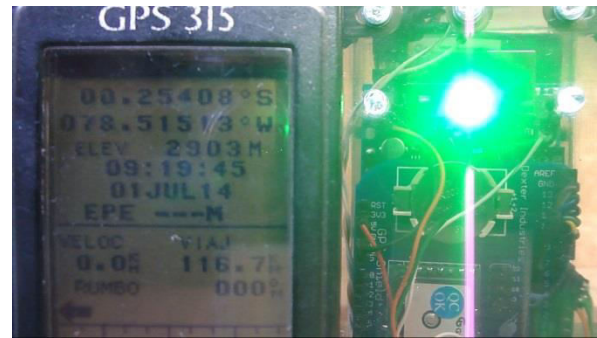
Dato 5



Dato 6



Dato 7



Dato 8



Dato 9



## Anexo E

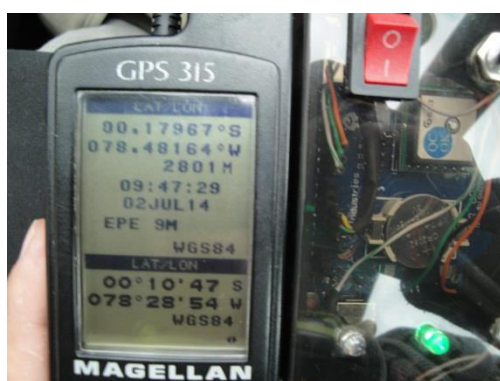
## Datos del GPS Magellan 315 Experimento 2, Ruta 1



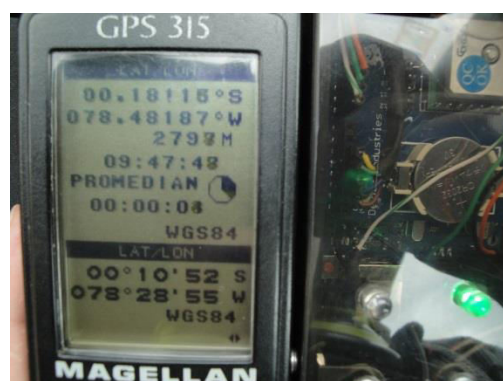
Dato 1



Dato 2



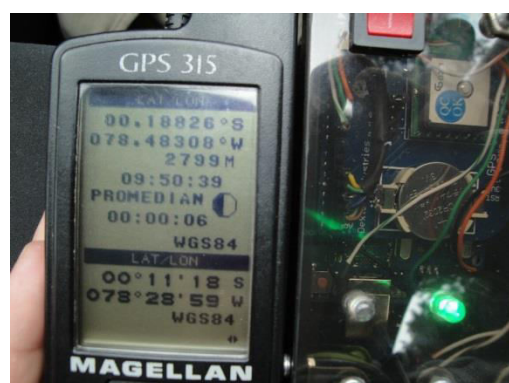
Dato 3



Dato 4



Dato 5



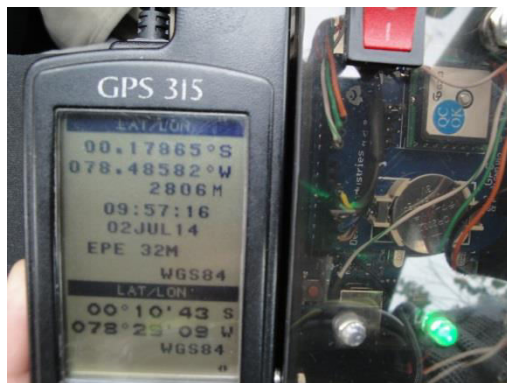
Dato 6



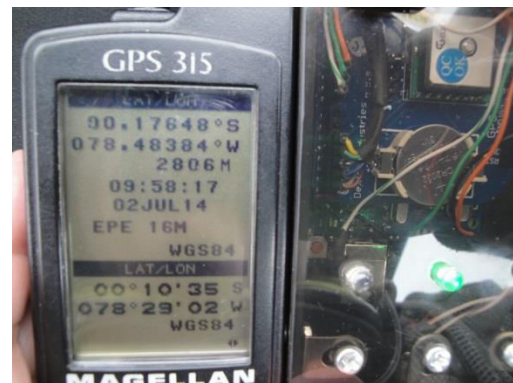
Dato 7



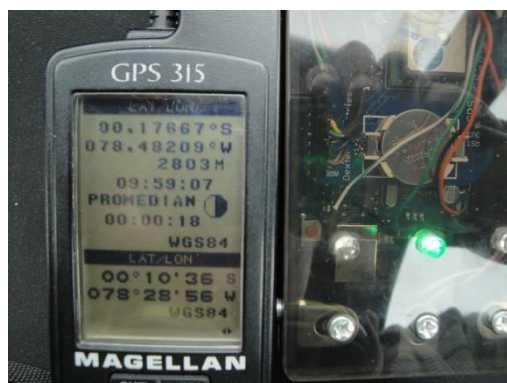
Dato 8



Dato 9



Dato 10

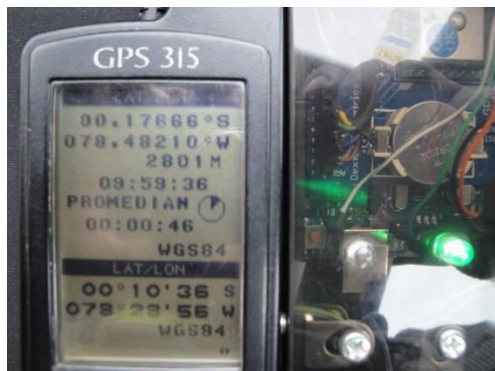


Dato 11



## Anexo F

## Datos del GPS Magellan 315 Experimento 2, Ruta 2



Dato 1



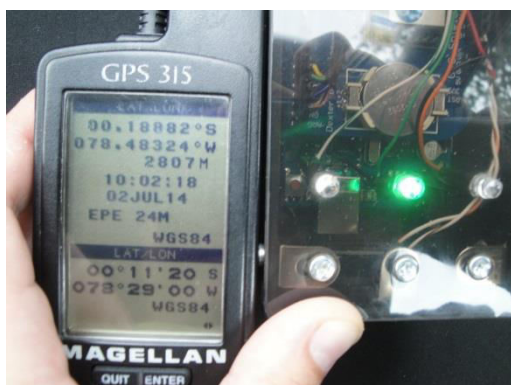
Dato 2



Dato 3



Dato 4



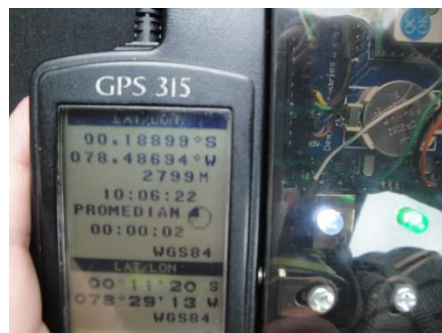
Dato 5



Dato 6



Dato 7



Dato 8



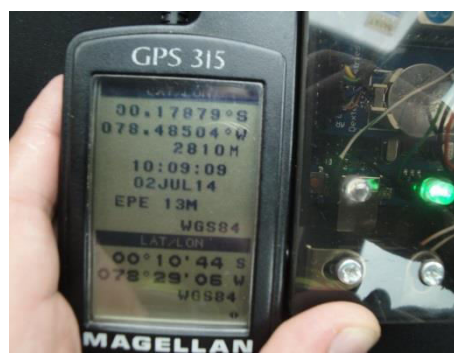
Dato 9



Dato 10



Dato 11



Dato 12



Dato 13



Dato 14

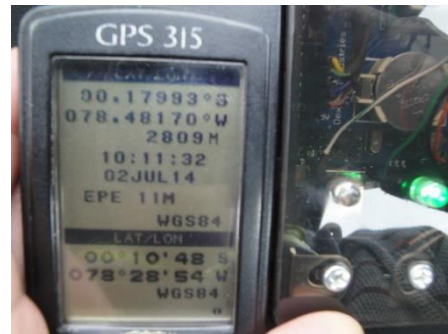


## Anexo G

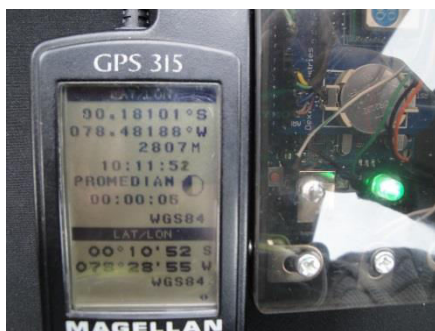
## Datos del GPS Magellan 315 Experimento 2, Ruta 3



Dato 1



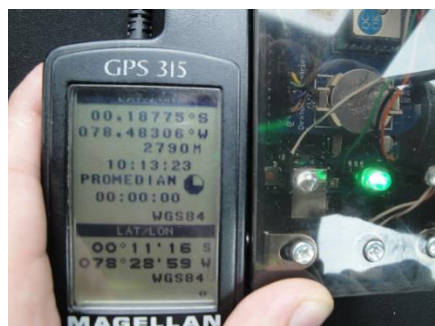
Dato 2



Dato 3



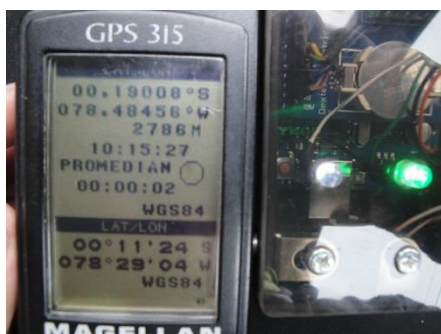
Dato 4



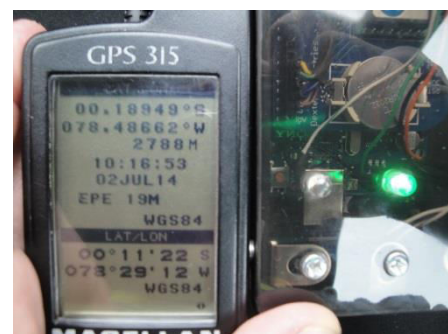
Dato 5



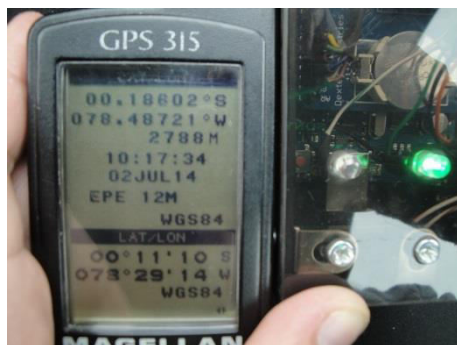
Dato 6



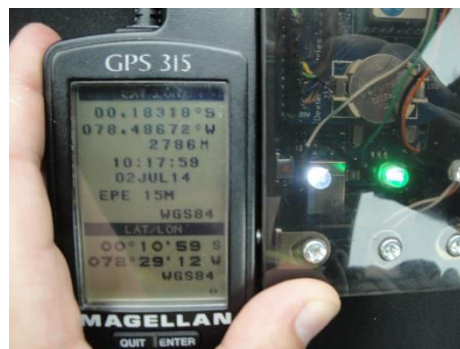
Dato 7



Dato 8



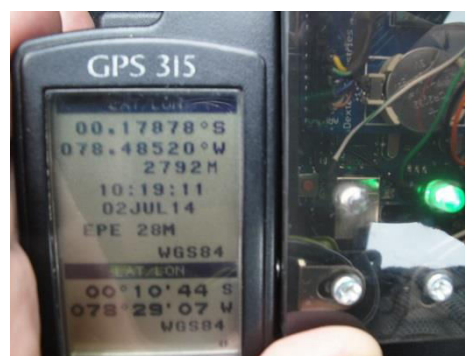
Dato 9



Dato 10



Dato 11



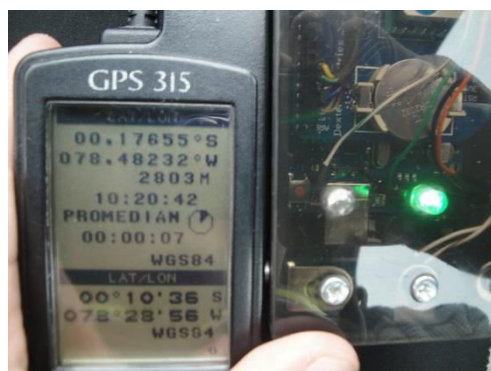
Dato 12



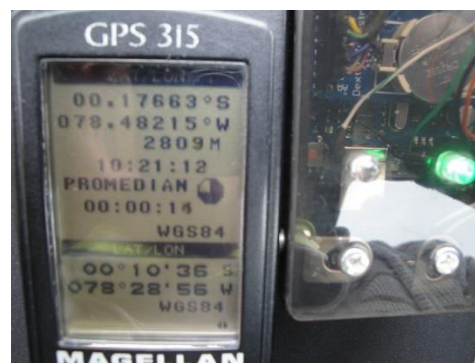
Dato 13



Dato 14



Dato 15



Dato 16