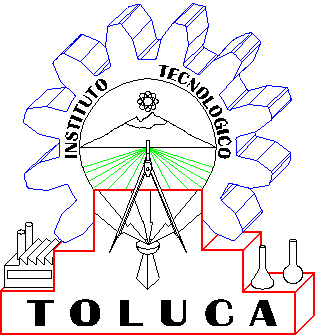
**SEP TECNM**



**INSTITUTO TECNOLÓGICO DE TOLUCA**

PROYECTO DE RESIDENCIA

**DISEÑO E IMPLEMENTACIÓN TECNOLÓGICA**

**CON MICROPROCESADORES INTEL GALILEO**

**INGENIERIA MECATRONICA**

**PRESENTAN:**

**Axel Shamir Favela Gómez 10280676**

**Benjamín Pineda Serrano 10280717**

**ASESOR INTERNO:**

**Ing. Alejandro Arellano Torres**

**ASESOR EXTERNO:**

**Ing. Manuel Haro Márquez**

Metepec, Estado de México, 9 de Junio del 2015

**ÍNDICE DE CONTENIDO**

[INTRODUCCIÓN 1](#_Toc421698914)

[JUSTIFICACÍON 3](#_Toc421698915)

[OBJETIVOS 4](#_Toc421698916)

[Objetivo General. 4](#_Toc421698917)

[Objetivos Particulares: 4](#_Toc421698918)

[CARACTERIZACIÓN DEL ÁREA EN QUE SE PARTICIPÓ 5](#_Toc421698919)

[PROBLEMAS A RESOLVER 6](#_Toc421698920)

[LIMITACIONES 6](#_Toc421698921)

[ALCANCE 7](#_Toc421698922)

[CAPÍTULO 1. ANTECEDENTES 8](#_Toc421698923)

[1.1 GPS 9](#_Toc421698924)

[1.2 Paneles Solares 12](#_Toc421698925)

[CAPÍTULO 2. MARCO TEÓRICO 15](#_Toc421698926)

[2.1 Tarjeta Intel Galileo 16](#_Toc421698927)

[2.1.1 Características de la tarjeta 17](#_Toc421698928)

[2.2 Arduino GPS Shield ITEA STUDIO 1.1 20](#_Toc421698929)

[2.2.1 Especificaciones 20](#_Toc421698930)

[CAPÍTULO 3. DESARROLLO 21](#_Toc421698931)

[3.1 Construcción del GPS 22](#_Toc421698932)

[3.2 Construcción del Cargador con Paneles Solares 29](#_Toc421698933)

[CAPÍTULO 4. DESCRIPCIÓN DETALLADA DE LAS ACTIVIDADES 32](#_Toc421698934)

[CAPÍTULO 5. RESULTADOS Y CONCLUSIONES 35](#_Toc421698935)

[5.1 RESULTADOS 36](#_Toc421698936)

[5.2 CONCLUSIONES 37](#_Toc421698937)

[REFERENCIAS 38](#_Toc421698938)

[ANEXOS 39](#_Toc421698939)

[Código a utilizar para el GPS sin pantalla. 40](#_Toc421698940)

[Código a utilizar para el GPS con la pantalla Display. 43](#_Toc421698941)

# INTRODUCCIÓN

El siguiente trabajo describe el proyecto que se desarrollará dentro de la organización conocida como Laboratorio de Desarrollo de Software Libre (LABSOL) ubicada en Zacatecas, Zacatecas; la cual depende del Consejo Zacatecano de Ciencia y Tecnología (COZCyT), en el área de desarrollo e innovación tecnológica.

Siempre ha sido un obstáculo para los deportistas el hecho de quedarse sin batería mientras realizan sus actividades, así como también han ocurrido sucesos en donde no se sabe en qué ubicación se encuentran en un terreno desconocido; es por eso que se han desarrollado distintos dispositivos que resuelvan estos problemas como lo son mochilas con paneles solares que almacenan la energía del sol para posteriormente utilizarla para recargar algún dispositivo móvil; o mochilas que cuentan con un sistema GPS para auxiliarles con la exploración y navegación dentro de una zona requerida.

Pero por si eso fuera poco también se han desarrollado bicicletas con la capacidad de entrar perfectamente en una mochila para facilitar su uso y su transporte en terrenos de difícil acceso para una de estas y si eso no es suficiente incluso se han creado bicicletas plegables que se convierten en mochilas, siendo aún más practico el ir a explorar con una bicicleta.

A pesar de todas estas innovaciones aún no ha surgido la tecnología que sea capaz de combinar todas las anteriores para formar así una “súper” mochila integrada que sirva para todos los objetivos mencionados. Es por eso que se ha decidido el implementar una mochila que tenga la seguridad de un dispositivo GPS, la practicidad de un cargador por medio de paneles solares y además se pueda convertir en una bicicleta plegable para trasportarse de manera ergonómica y económica.

Para lograr dicho invento será necesario la utilización de varios componentes eléctricos como los paneles solares, capacitores, conectores, entre otros; pero lo más importante para el funcionamiento GPS será el uso de un microprocesador Intel Galileo, el cual se encargará de ser el cerebro del sistema.

# JUSTIFICACÍON

Existe una simplicidad en los proyectos relacionados con facilitar el trayecto de los deportistas y excursionistas, ya que estos solamente se centran a una sola necesidad, haciendo que sea más difícil el llevarlos durante la travesía, pues ocupan un gran espacio y peso.

Es por eso que se ha decidido crear un prototipo en donde se haga una combinación de estas tecnologías mediante la implementación de un microprocesador Intel Galileo para el correcto control de cada una de las funciones, agregando la opción de convertirse en una bicicleta plegable capaz de transitar por terrenos difíciles, de esta manera se facilitaría su uso, ya que el espacio utilizado por este no sería mayor al de una mochila comercial, en donde sus materiales de construcción serian resistentes pero de bajo peso, contaría con las funciones de un localizador satelital GPS para los casos en que sea desconocida la ubicación y un dispositivo capaz de cargar dispositivos móviles a base de energía solar, de esta manera se obtendría un proyecto mucho más íntegro, innovador y de bajo costo.

# OBJETIVOS

## Objetivo General.

Desarrollar un dispositivo por medio de un Intel Galileo que sea capaz de usarse como mochila, que pueda localizarse por medio de GPS, que cuente con un circuito que tenga la capacidad de almacenar la energía por medio de paneles solares para que después sea usada para cargar un dispositivo móvil y además este adaptado a una bicicleta plegable.

## Objetivos Particulares:

* Investigar sobre todos los componentes que se necesiten para desarrollar el prototipo.
* Diseñar el circuito de comunicación con el GPS.
* Desarrollar el circuito de almacenamiento de energía con los paneles solares.
* Adaptar una mochila a la bicicleta plegable.
* Combinar todo lo anterior para desarrollar un prototipo.
* Efectuar las pruebas y mejorar según se necesite.
* Documentar cada una de las fases.

# CARACTERIZACIÓN DEL ÁREA EN QUE SE PARTICIPÓ

**Nombre:** Laboratorio de Desarrollo de Software Libre (LABSOL).

**Domicilio:** Calle De la Juventud 504 Col. Barros Sierra.

**Giro:** Servicios.

**Representante:** Ing. Manuel Haro Márquez.

**Área y número telefónico donde se realizara el proyecto:** LABSOL,

Teléfono: (52-492) 9212816.

**Aportación:** Desarrollo e Innovación Tecnológica.

El Laboratorio de Desarrollo de Software Libre (LABSOL) se crea con la intención de ir conformando una comunidad activa hacia todos los sectores productivos, teniendo como objetivo principal el impulsar el desarrollo científico y tecnológico mediante el estímulo a la investigación y a la creación de redes de vinculación productiva, así también como la difusión y la divulgación de la ciencia y sus distintas aplicaciones en la industria, para así evaluar la cultura científica de la sociedad y favorecer su mejoramiento integral.

Se compromete a impulsar el desarrollo de sus líneas de acción, fortaleciendo sus estrategias con proyectos innovadores, haciendo uso de la aplicación de las tecnologías y sus desarrollos científicos.

El área en que se trabajó se encarga del desarrollo de proyectos integrados, desarrollados por un equipo de trabajo comprendido por miembros de distintas aéreas, uniendo las partes de software y hardware de manera que dichos proyectos tengan un correcto funcionamiento y aplicación dentro de la industria y el mercado actual, con el fin de que los conocimientos desarrollados con estos puedan ser retomados y mejorados por otros miembros del equipo o por futuros desarrolladores.

# PROBLEMAS A RESOLVER

1. Obtener el material.
2. Lograr la compatibilidad entre el Arduino Intel Galileo y el GPS Shield.
3. Diseñar el programa del Intel Galileo para el funcionamiento del GPS.
4. Desarrollar el circuito de carga con los paneles solares.
5. Adaptar el diseño de la bicicleta plegable.
6. Modelar el prototipo conjunto.
7. Crear el prototipo.

# LIMITACIONES

1. El proyecto se realizara en un periodo de tiempo de cuatro a cinco meses a partir de la fecha señalada.
2. Debido a la cantidad de piezas y componentes electrónicos y mecánicos que se utilizaran en el diseño e implementación del prototipo, es posible que no se logre realizar de la manera deseada cada una de las actividades establecidas.

# ALCANCE

La implementación de este prototipo, está diseñado con la capacidad de poder localizar la posición exacta del mismo en cualquier lugar, mediante el uso de un GPS, también contara con un cargador solar para dispositivos móviles de bajo consumo, además de tener la capacidad de desplegar una bicicleta para su uso en el traslado de pequeñas y medianas distancias.

Se contara con un diseño ergonómico capaz de adaptarse adecuadamente al usuario y al terreno, teniendo un tamaño no mayor a las mochilas escolares comerciales. Este está enfocado en ayudar a los deportistas dentro de las dificultades a las que se pueden encontrar en un área desconocida sin las herramientas necesarias para la supervivencia.

Se brindara información detallada del funcionamiento, así como la forma de utilizarlo correctamente.

# CAPÍTULO 1. ANTECEDENTES

## 1.1 GPS

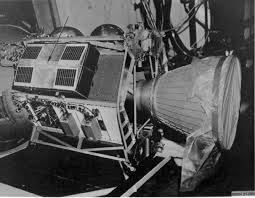
El primer sistema de navegación basado en satélites apareció en 1965. Al principio de los 60 los departamentos de defensa, transporte y la agencia espacial norteamericanas (DoD, DoT y NASA respectivamente) tomaron intereses para desarrollar un sistema para determinar la posición basado en satélites.

El sistema debía cumplir los siguientes requisitos:

* **Globalidad.-** Que abarcara toda la superficie del globo.
* **Continuidad.-** Que tuviera un funcionamiento continuo sin afectarle las condiciones atmosféricas.
* **Altamente Dinámico.-** Que posibilitara su uso en aviación y precisión.

Esto llevo a producir diferentes experimentos como el Timation y el sistema 621B en desiertos simulando diferentes comportamientos.

El sistema TRANSIT estaba constituido por una constelación de seis satélites en órbita polar baja, a una altura de 1074 Km (Figura 1). Tal configuración conseguía una cobertura mundial pero no constante. La posibilidad de posicionarse era intermitente, pudiéndose acceder a los satélites cada 1.5 h. El cálculo de la posición requería estar siguiendo al satélite durante quince minutos continuamente.



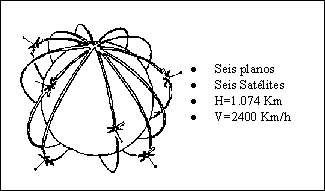


Figura 1: GPS y ejemplo gráfico de su posicionamiento en la atmósfera.

TRANSIT trabajaba con dos señales en dos frecuencias, para evitar los errores debidos a la perturbación ionosférica. El cálculo de la posición se basaba en la medida continua de la desviación de frecuencia Doppler de la señal recibida y su posterior comparación con tablas y gráficos. El error de TRANSIT estaba en torno a los 250 m. Su gran aplicación fue la navegación de submarinos y de barcos.

Durante la guerra fría se concibió un sistema formado por 24 satélites en órbita media, que diera cobertura global y continua (ver figura 2).

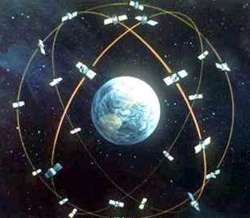


Figura 2: Acoplamiento de los 24 satélites al GPS.

El primer satélite se lanzó en 1978, y se planificó tener la constelación completa ocho años después. Por fin, en diciembre de 1983 se declaró la fase operativa inicial del sistema GPS. El objetivo del sistema GPS era ofrecer a las fuerzas de los EE.UU. la posibilidad de posicionarse (disponer de la posición geográfica) de forma autónoma o individual, de vehículos o armamento, con un coste relativamente bajo, con disponibilidad global y sin restricciones temporales.

En 1984  un vuelo civil de Korean Airlines fue derribado por la Unión Soviética al invadir por error su espacio aéreo. Ello llevó a la administración Reagan a ofrecer a los usuarios civiles cierto nivel de uso de GPS, llegando finalmente a ceder el uso global y sin restricciones temporales, de esta forma se conseguía un retorno a la economía de los EE.UU. inimaginables unos años atrás. Además suponía un gran liderazgo tecnológico originando un vertiginoso mercado de aplicaciones (Figura 3).



Figura 3: Ejemplos de aplicaciones civiles para el GPS.

Desde 1984, con muy pocos satélites en órbita, aparecieron tímidamente fabricantes de receptores GPS destinados al mundo civil como se muestra en la Figura 4.



Figura 4: Compañías usuarias del GPS.

Los datos que recibiremos en nuestro modulo GPS siguen el protocolo NMEA siglas de National Marine Electronics Asociation.  Tienen esta estructura:

**$GPRMC,044235.000,A,4322.0289,N,00824.5210,W,0.39,65.46,020911,,,A\*44**

Empiezan por "$" y acaban en un "A\*" seguido de dos números, éste es el checksum para poder saber si el dato recibido es correcto. Los datos se separan por comas y el primero es el tipo de transmisión, en este caso el GPRMC (o RMC), uno de los más usados, que por ejemplo no incluye el valor de altitud, que si se incluye en el GPGGA. Todo el protocolo lo enlazo en la sección de descargas. Revisemos los datos para entenderlos:

* **044235.000**es la hora GMT (04:42:35)
* **A**es la indicación de que el dato de posición está fijado y es correcto. V seria no válido
* **4322.0289**es la longitud (43º 22.0289´)
* **N**Norte
* **00824.5210**es la latitud (8º 24.5210´)
* **W** Oeste
* **0.39** velocidad en nudos
* **65.46** orientación en grados
* **020911** fecha (2 de septiembre del 2011)

## 1.2 Paneles Solares

El uso de la energía solar siempre ha sido un recurso importante desde hace más de 100 años, ya que ha resultado ser una fuente productiva y renovable de energía para diferentes usos como lo es la producción de vapor para hornos o calefacción.

Pero no fue hasta el descubrimiento del científico francés Alexandre Edmon Becquerel (Figura 5), que al estar experimentando con una pila electrolítica sumergida en una sustancia de las mismas propiedades, observó que después al exponerla a la luz generaba más electricidad, a lo cual se le llamó después como “Efecto Fotovoltaico” en 1839 que consiste en la conversión de la luz del sol en energía eléctrica.

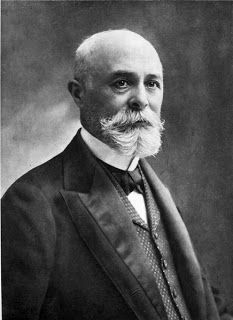


Figura 5: Retrato de Alexandre Edmon Becquerel

En 1885 el profesor W. Grylls Adams experimentó con el selenio y nombró como corriente fotoeléctrica al flujo de corriente que se generaba de la reacción del silicio con la luz.

EL descubrimiento de Becquerel llevó a la invención en 1893 por Charles Fritts de la primera célula solar real que fue formada por láminas de revestimiento de selenio con una fina capa de oro (véase la Figura 6). Y desde este humilde comienzo se plantearía el dispositivo que hoy conocemos como el panel solar.

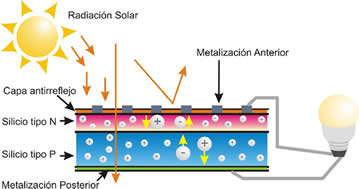


Figura 6: Composición de la primera Célula Solar.

El inventor Russel Ohl patentó las primeras células solares de silicio en 1946, pero Gerald Paterson de Laboratorios Bell creó una célula fotovoltaica más eficiente y gracias a esto Daryl Chaplin y Calvin Fuller mejoraron estas células solares para un uso más práctico.

Esto llevo a la producción de paneles solares por la misma empresa, utilizados en su mayoría satélites espaciales. En la década de los 70 se creó el primer uso público para los paneles solares, el cual fue una calculadora como las que actualmente se utilizan (ver Figura 7).

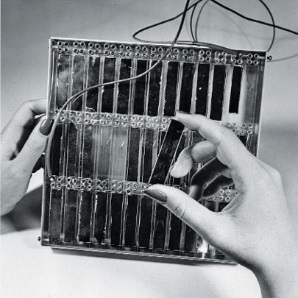
 

Figura 7: Ejemplos de calculadoras que utilizan paneles solares.

Hoy en día, los paneles solares y sistemas completos de paneles solares se utilizan para una amplia variedad de aplicaciones, como por ejemplo la alimentación eléctrica de casas o edificios comerciales como la sede de Google California, como se muestra en la Figura 8.



Figura 8: Instalación de Paneles Solares de Google.

A pesar de todo el desarrollo tecnológico aún no ha parecido la tecnología que combine la funcionalidad de los paneles solares con un dispositivo geolocalizador con el objetivo de volver las exploraciones más confortables y sencillas.

# CAPÍTULO 2. MARCO TEÓRICO

## 2.1 Tarjeta Intel Galileo

En el 2013 Intel presento en el “maker fire” en Roma un nuevo acuerdo de colaboración con Arduino LLC que dio como fruto la tarjeta Intel Galileo, la cual ofrece las mismas posibilidades que otras tarjetas como Raspberry Pi o Arduino, es una plataforma perfecta para artistas, diseñadores e inventores en la cual se pueden crear infinidad de proyectos (véase Figura 9).

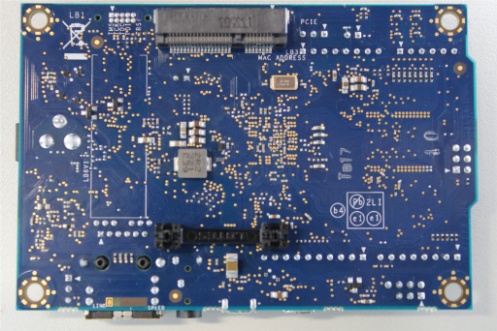
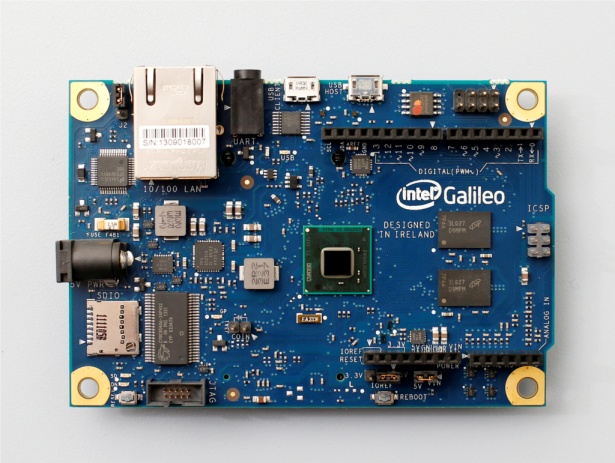


Figura 9: Tarjeta Intel Galileo.

Intel Galileo es la primera de una línea de tarjetas de desarrollo certificadas por Arduino basadas en la arquitectura Intel x86 y está diseñado para las comunidades fabricante y educación. La tarjeta de desarrollo se ejecuta en un sistema operativo Linux de código abierto con las bibliotecas de software de Arduino, lo cual permite la reutilización de algún software existente. Intel Galileo se puede programar a través de OS X, Microsoft Windows y software operativo host Linux.

Es una tarjeta electrónica basada en el procesador Intel® Quark SoC X1000 aplicación, un sistema de clase Pentium Intel de 32 bits en un chip. La tarjeta también está diseñada para ser el hardware y el software compatible con los “shields” de una placa Arduino. Pines digitales 0 a 13 (y la AREF adyacente y pines GND), entradas analógicas de 0 a 5, el jefe de la energía, de cabecera ICSP, y los pines del puerto UART (0 y 1), están todos en los mismos lugares que en la Arduino Uno R3. Esto también se conoce como el 1,0 pinout Arduino.

### 2.1.1 Características de la tarjeta

Galileo está diseñado para soportar “shields” que operan en cualquiera de 3.3V o 5V. El voltaje de funcionamiento básico de Galileo es 3.3V. Sin embargo, un puente en la tarjeta permite el cambio de voltaje a 5V a los pines de E / S. Esto proporciona soporte para “shields” Uno 5V y se convierte en el comportamiento predeterminado. Al cambiar la posición del puente, la conversión de voltaje, se puede desactivar para proporcionar un funcionamiento 3.3V en los pines de E/S.

Además de la compatibilidad de software y hardware con Arduino, la placa Galileo tiene varios estándar de la industria de PC puertos E / S y características para ampliar el uso y capacidades nativas más allá de los “shields” Arduino. ACPI, PCI Express, Ethernet 10/100 Mb, lector SD, puertos para dispositivos USB 2.0 y EHCI/OHCI USB, UART de alta velocidad, puerto serie RS-232, flash NOR programable de 8 MB y un puerto JTAG para mayor comodidad en la depuración de software.

Descripción de los elementos que conformar la placa Intel Galileo:

Bus I2C, TWI: Es un protocolo de comunicación serie, SDA transmite los datos y SCL es la señal de reloj, estos pines están cerca al pin AREF.

TWI: El pin A4 o A5 o SDA y SCL pin. Soporta comunicación TWI utilizando la librería “Wire”.

SPI (*Serial Peripheral Interface*): El valor predeterminado es de 4 MHz para soportar “shields” Arduino Uno.

Entradas / Salidas: 14 pines de entrada / salida digitales, 6 de estas pueden utilizarse para salidas de modulación de ancho de pulso (PWM); Cada uno de los 14 pines digitales en Galileo se puede utilizar como una entrada o salida, utilizando “pinMode”, “digitalWrite”, y las funciones “digitalRead”. Operan en 3,3 o 5 voltios. Cada pin puede proporcionar un máximo de 10 mA o recibir un máximo de 25 mA y tiene una resistencia de pull-up (desconectado por defecto) de 5,6 K a 10 kOhm.

A0 - A5: 6 entradas analógicas, a través de un convertidor AD7298-A a D. Cada una de las 6 entradas analógicas, etiquetados A0 a A5, ofrece 12 bits de resolución (es decir, 4.096 valores diferentes). Por defecto se miden desde el suelo a 5 voltios.

**Nota:** Si bien Galileo tiene un controlador nativo SPI, actuará como maestro y no como un esclavo SPI. Por lo tanto, Galileo no puede ser esclavo SPI a otro maestro SPI. Puede actuar, sin embargo, como un dispositivo esclavo a través del conector de cliente USB.

UART (puerto serie): Puertos de velocidad programable UART (pines digitales 0 (RX) y 1 (TX))

ICSP (SPI): Situado adecuadamente para conectar los “shields” existentes. Estos pines soportan la comunicación SPI utilizando la librería SPI.

VIN: La tarjeta necesita una fuente de alimentación externa de 5V conectada al “power jack”, se puede tener este voltaje a través de este pin o también puede remplazarse el voltaje del “power jack” y alimentarlo por este pin.

Pin de salida de 5V: Este pin da salida 5V de la fuente externa o del conector USB. Consumo de corriente máxima es de 800 mA

Pin de salida 3.3V: Un suministro de 3,3 voltios generada por el regulador de la tarjeta. Consumo de corriente máxima es de 800 Ma

Pines de tierra: GND.

IOREF: Este es el voltaje correspondiente a las E / S de la placa, un puente de selección en la tarjeta se utiliza para seleccionar entre 3.3V y 5V operación de “shields”.

Botón RESET: Poner este botón en bajo para reiniciar la tarjeta.

A continuación se muestra en la Figura 10 la composición electrónica de la tarjeta:

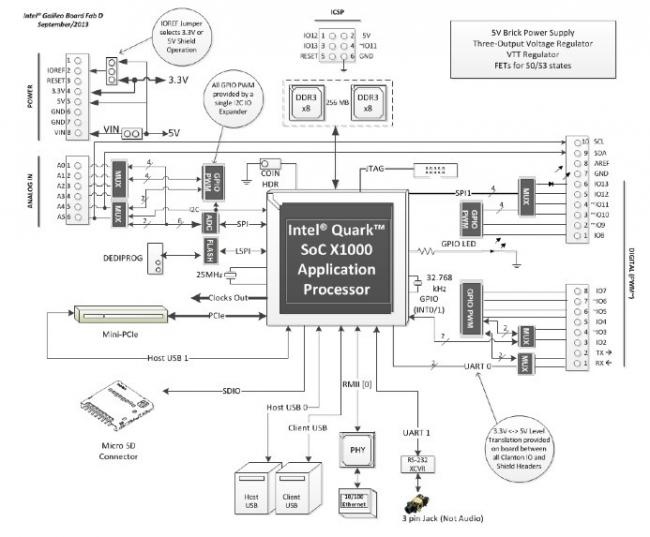


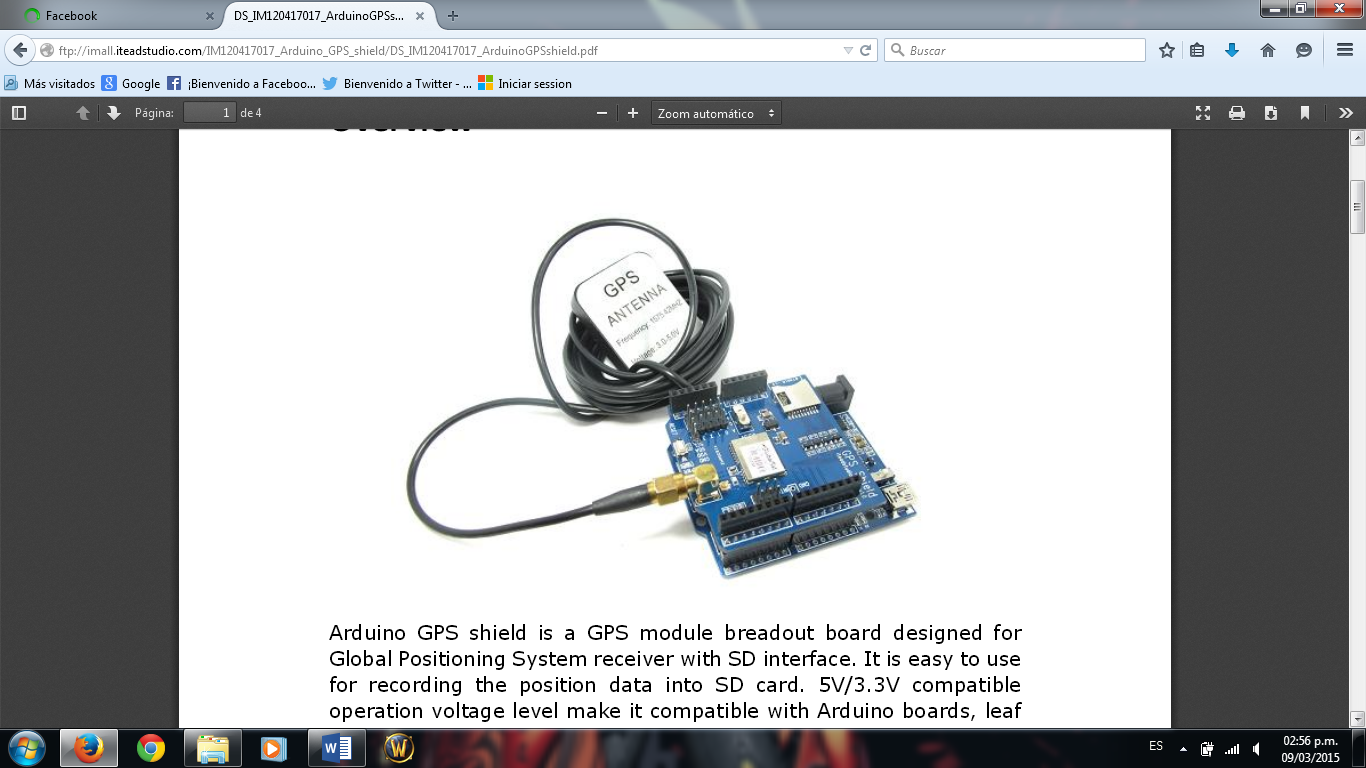
Figura 10: Diagrama del Intel Galileo.

Como se puede ver, estamos ante un verdadero centro neurálgico para gobernar cualquier proyecto que un desarrollador tenga en mente.

## 2.2 Arduino GPS Shield ITEA STUDIO 1.1

El Arduino GPS shield es un módulo GPS diseñado para la recepción de datos del sistema de posicionamiento global con una interfaz SD (ver Figura 11). Es fácil de usar para el registro de los datos de posición en la tarjeta SD.

Es compatible al nivel de tensión de operación de 5V / 3.3V, los que lo hacen que sea compatible con la placa Arduino, IFlat32 y otras placas compatibles con arduino.



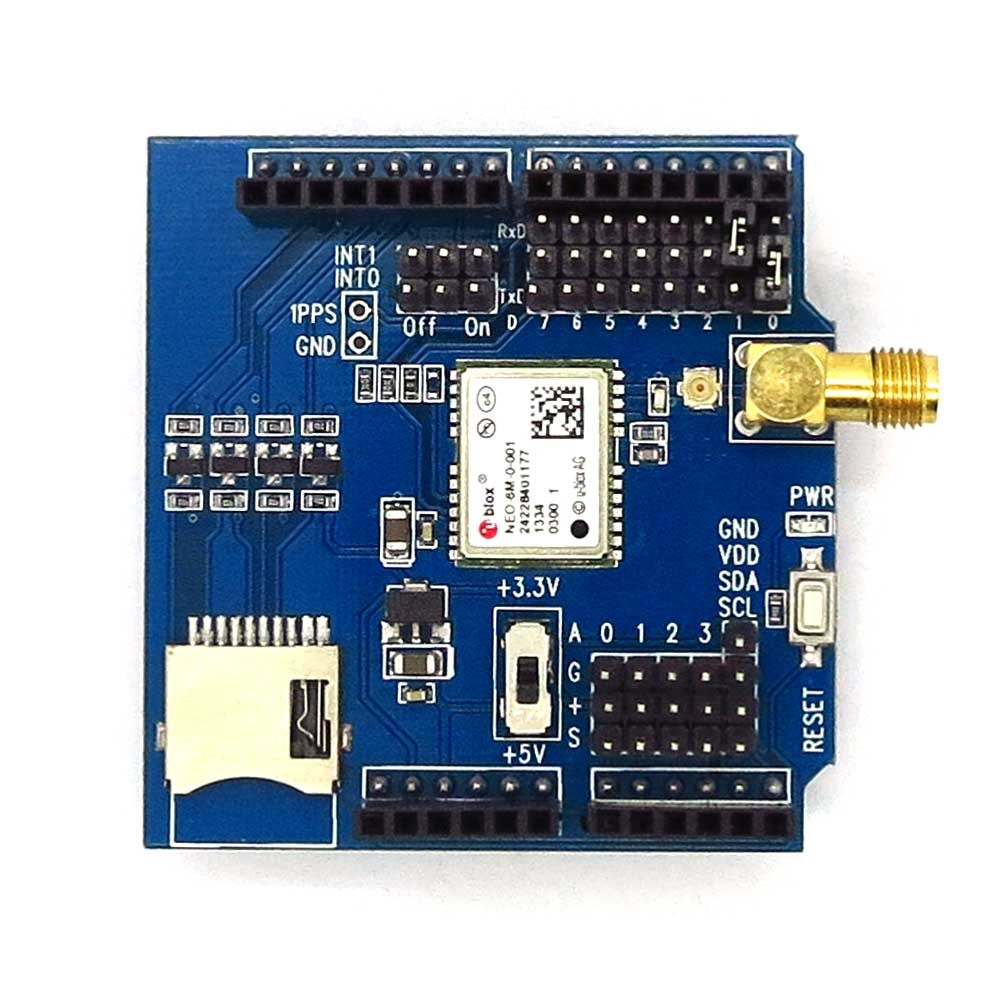


Figura 11: Módulo GPS para Arduino.

### Especificaciones

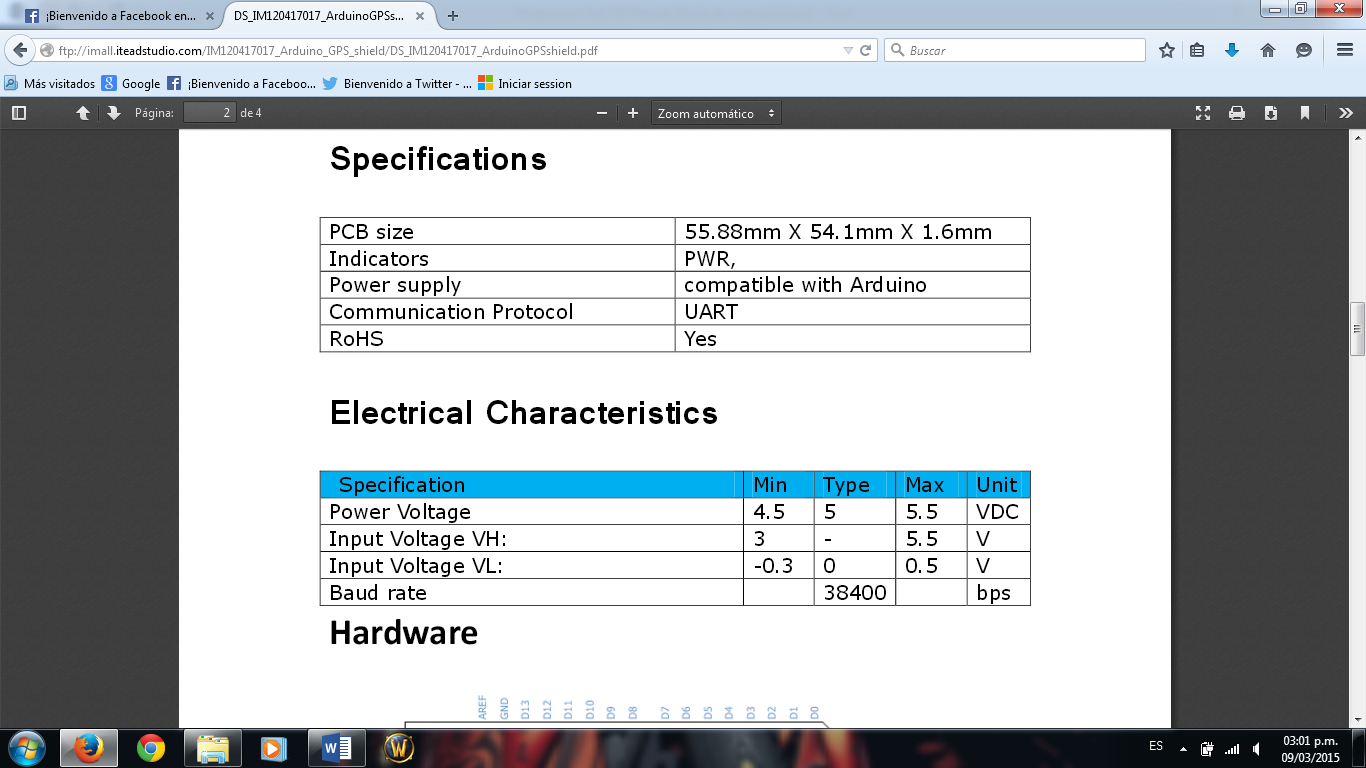


Tabla 1: Especificaciones Físicas del Módulo.

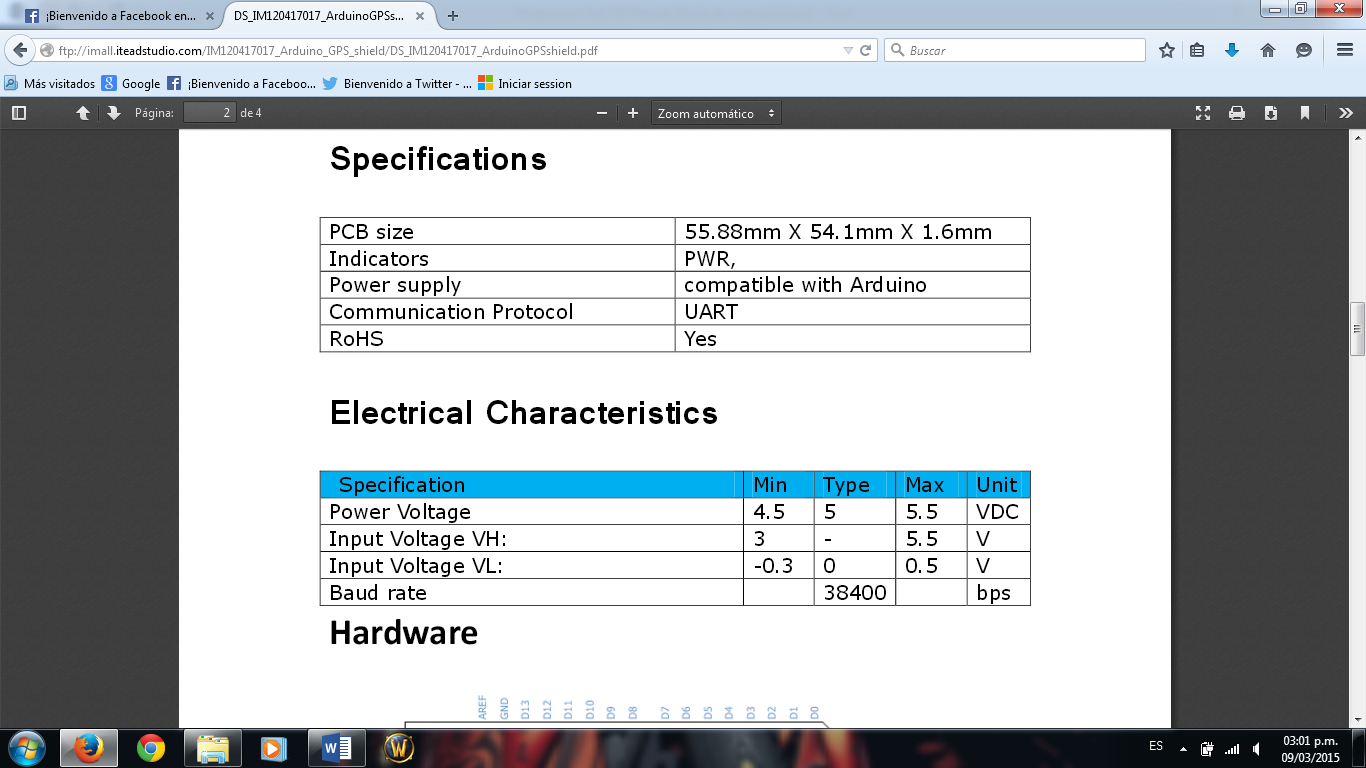


Tabla 2: Especificaciones Eléctricas.

# CAPÍTULO 3. DESARROLLO

Para la construcción del proyecto serán necesarios los siguientes componentes:

* 1. Intel Galileo
  2. Módulo GPS ITEAD Studio 1.1
  3. Antena para GPS
  4. Cable USB
  5. Micro SD
  6. Paneles Solares
  7. Elementos electrónicos básicos (resistencias, capacitores, etc)

## 3.1 Construcción del GPS

Alimentamos al Intel galileo por medio de su conector para la fuente externa y posteriormente lo conectamos a la computadora por medio del cable USB para así poder introducir el programa.

Una vez que tenemos al Intel galileo programado, debemos desconectarlo de la alimentación para añadir los demás componentes que se tienen que conectar, como lo es el módulo de GPS.

Al contar con el modulo solamente es necesario colocarlo sobre la placa del galileo asegurándonos que el pin 0 del módulo entre con el pin 0 de los pines digitales del Intel galileo.

Ya con todo conectado alimentamos de nuevo al Intel galileo como lo hicimos anteriormente y comenzamos a captar la señal, será necesario salir y colocar la antena en un punto donde tenga vista directa hacia el cielo, sin olvidarse de introducir la tarjeta micro SD en la ranura del módulo para poder guardar la información.

Después de haber obtenido la información extraemos el micro SD y la introducimos en la computadora para observar si guardo algún tipo de información, si lo hizo bien podremos encontrar un archivo .txt dentro de la memoria (ver figura 12).



Figura 12: Datos nmea capturados por el GPS.

Cerramos el archivo y le cambiamos la terminación .txt por la terminación .nmea para que así podamos obtener las locaciones que el gps haya captado en cualquier visualizador de datos nmea, por ejemplo, Google Earth.

Abrimos el programa y en la opción de herramientas seleccionamos GPS como se muestra en la Figura 13.

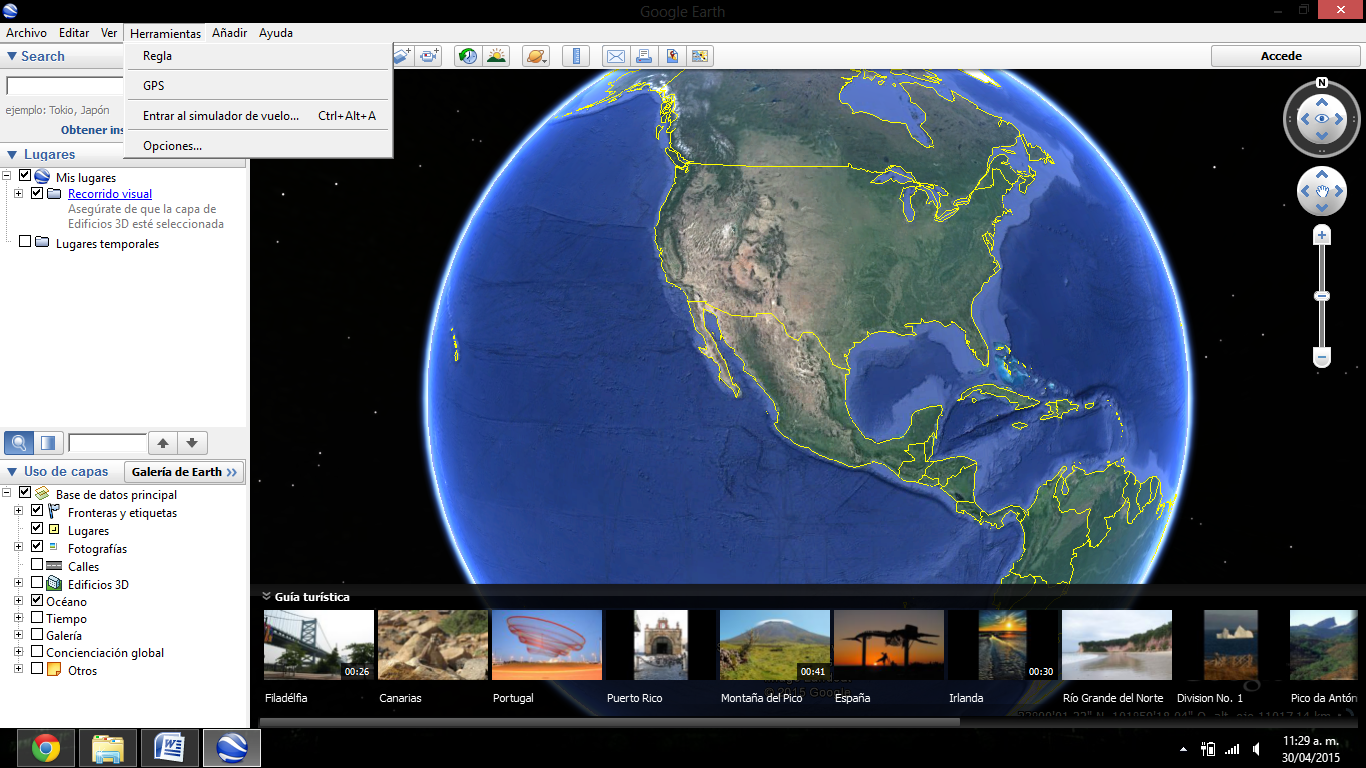


Figura 13: Abriendo la opción de GPS dentro de Google Earth.

Luego dentro del menú de gps elegimos la opción de importar archivo, activamos la opción de KML Linestrings para tener una mejor lectura de la información y le damos en importar.

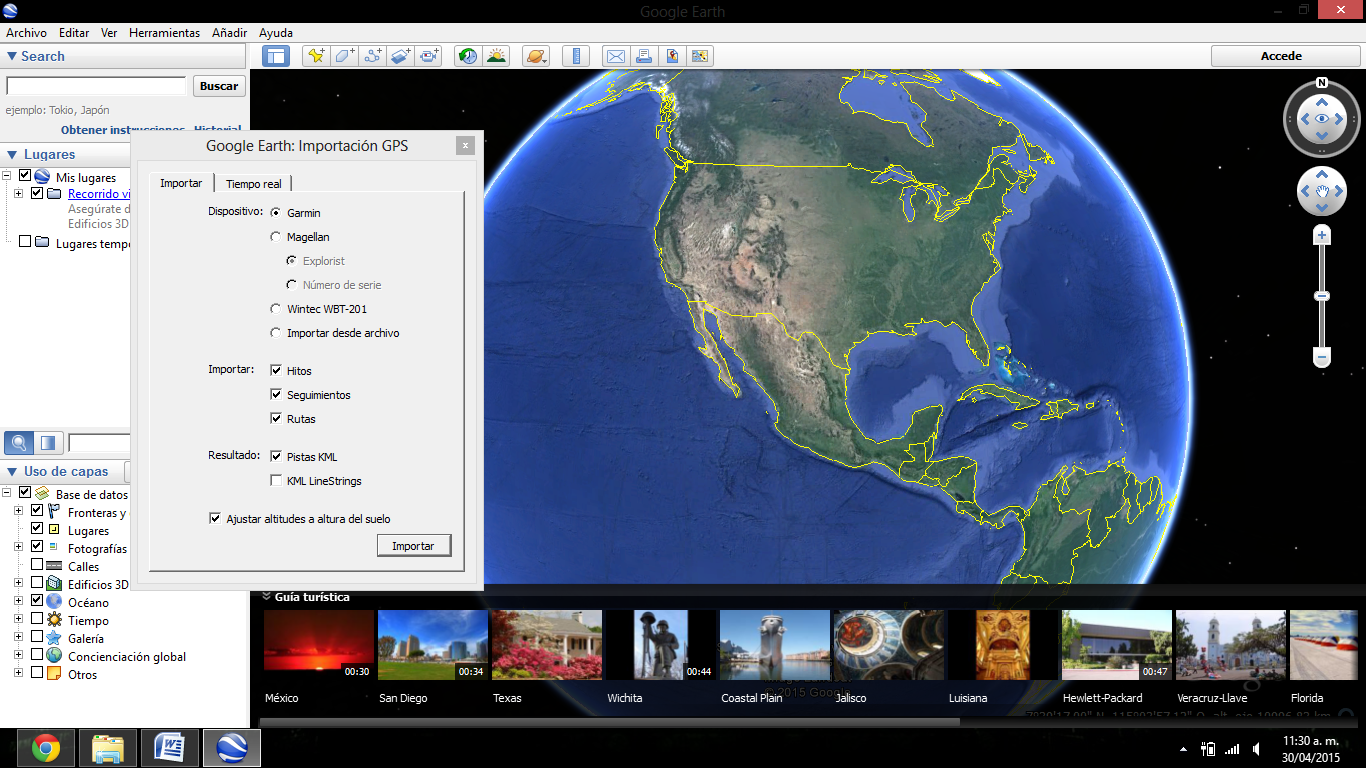


Figura 14: Importación de los datos nmea.

Y listo, ahora el programa nos mostrará las coordenadas que captó, como puede observar en la Figura 15:



Figura 15: Coordenadas mostradas en Google Earth.

Además de tener un micro SD para guardar los datos nmea, también podemos agregar una pantalla display para ir observando las coordenadas directamente, las conexiones se realizan como se presenta en la Tabla 3:

|  |  |
| --- | --- |
| LCD Pin | Conectado a |
| 1 (VSS) | GND Arduino pin \* |
| 2 (VDD) | + 5v pin de Arduino |
| 3 (contraste) | Resistencia de 2.2k a GND |
| 4 RS | Pin de Arduino 12 |
| 5 R / W | Pin de Arduino 11 |
| 6 Habilitar | Pin de Arduino 10 |
| 7 No hay conexión |  |
| 8 No hay conexión |  |
| 9 No hay conexión |  |
| 10 No hay conexión |  |
| 11 (datos 4) | Arduino pin 5 |
| 12 (Datos 5) | Pin de Arduino 4 |
| 13 (Data 6) | Arduino pin 3 |
| 14 (Data 7) | Arduino pin 2 |
| 15 Luz de fondo + | Resistencia de 1.5k para pin de Arduino 13 |
| 16 Luz de fondo GND | GND Arduino pin \* |

Tabla 3: Conexiones entre Arduino y Monitor Display.

El GPS cuenta con unos jumpers que deben de conectarse de acuerdo a la función que se le quiere dar, como queremos que transmita los datos al display es necesario conectarlos como se ejemplifica en la Figura 16.

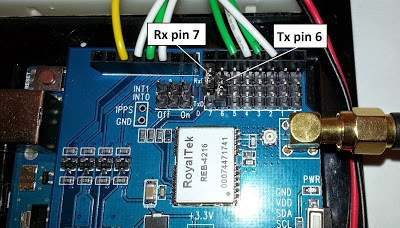


Figura 16: Jumpers conectados para la transmisión de datos.

Al haber realizado las conexiones correspondientes, se procede a subir el programa necesario para el funcionamiento del gps con visualización en la pantalla LCD, después de esto debe pasar un tiempo de entre 5 y 10 minutos para que la antena GPS capte la señal satelital y comience a mostrar las coordenadas (ver Figura 17).

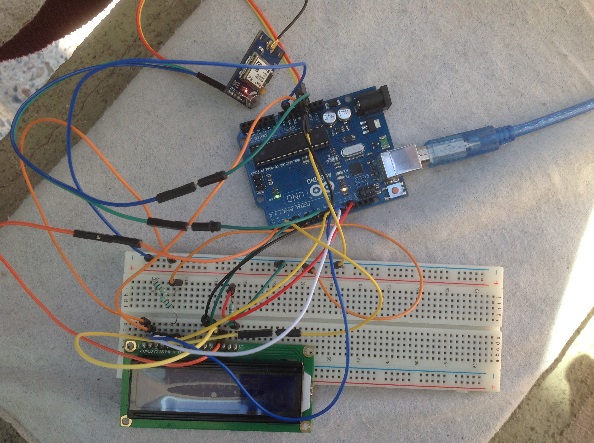
 

Figura 17: Circuito del GPS y Display funcionando.

Una vez terminado el circuito se hace el diseño para trasladarlo a una PCB para mejorar las conexiones, disminuir los cables y reducir el espacio (véase Figura 18).

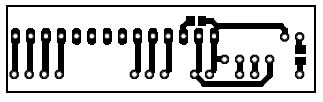
 

Figura 18: Diseño del circuito y de la PCB del GPS con Display en Eagle Professional.

Y si se desea también se pueden introducir las coordenadas nmea a cualquier otro programa de geolocalización como Google Maps (Figura 19).

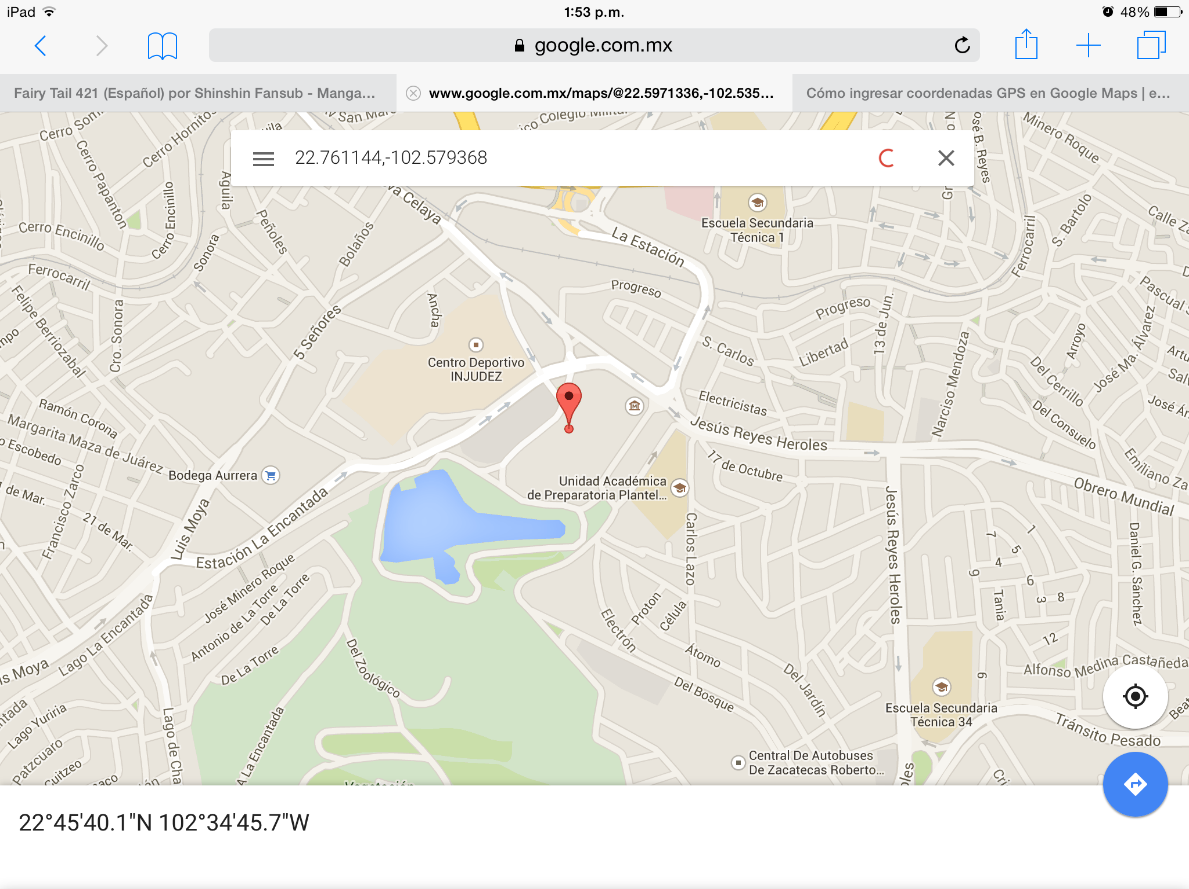


Figura 19: Coordenadas nmea introducidas en Google Maps.

## 3.2 Construcción del Cargador con Paneles Solares

Para la construcción del cargador necesitaremos de los siguientes materiales (Figura 20):

* Un cargador de dispositivos electrónicos con salida de 5v y conector de USB.
* Paneles solares suficientes para tener una corriente de 1 ampere.
* Una batería recargable que pueda dar 5v de salida.
* Cable, soldadura y componentes electrónicos básicos (resistencias).
* Un switch.





Figura 20: Ejemplo de panel solar y de cargador a utilizar.

El circuito que se usará cuenta con un divisor de voltaje para evitar voltajes mayores a los deseados, como se muestra en la Figura 21.

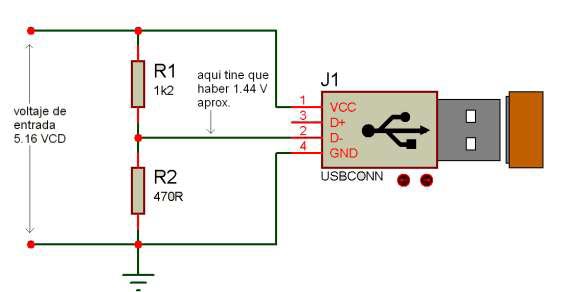


Figura 21: Divisor de voltaje para el conector USB

Comenzamos conectando todos los paneles en paralelo, es decir, tomamos el positivo del primer panel y lo conectamos al positivo del segundo panel, después tomamos los negativos y los conectamos y así sucesivamente dependiendo de la cantidad de paneles solares que se hayan decidido utilizar (ver Figura 22).



Figura 22: Paneles conectados en paralelo.

**NOTA:** En este caso se conectaron en paralelo debido a que se deseaba aumentar la corriente de salida para alimentar correctamente al cargador y así los componentes recibieran lo necesario; esto debido a que los paneles que utilizamos daban solamente 50mA cada uno.

Ya con los paneles conectados tomamos el positivo común y lo conectamos al switch, el switch nos servirá para elegir si queremos tomar la carga de los paneles o de la batería; y el negativo común lo enviamos a tierra. De la batería tomamos el positivo y lo mandamos de igual manera al switch y el negativo se envía a tierra. Después de las salidas del switch las mandamos hacia la entrada del cargador de dispositivos, después de tener todo conectado podemos usar cinta para tener los componentes fijos (véase Figura 23).

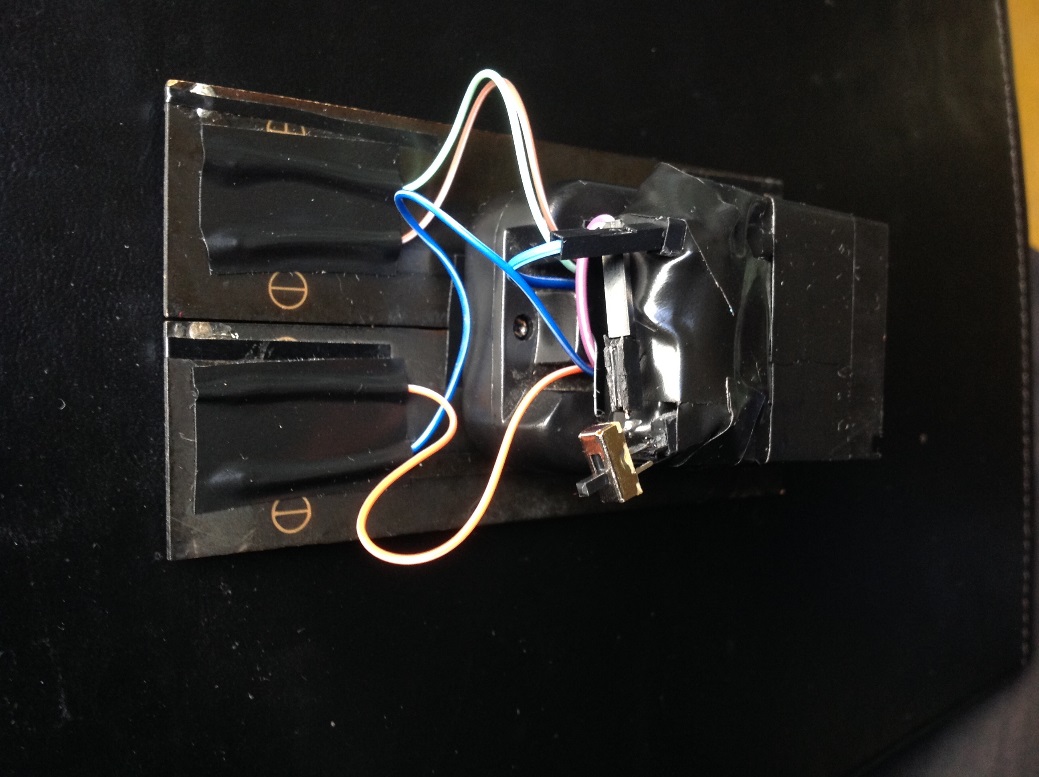


Figura 23: Cargador terminado.

Así como lo hicimos con el GPS, con este cargador debe ser igual, se diseña el circuito para optimizarlo, como se ve en la Figura 24.

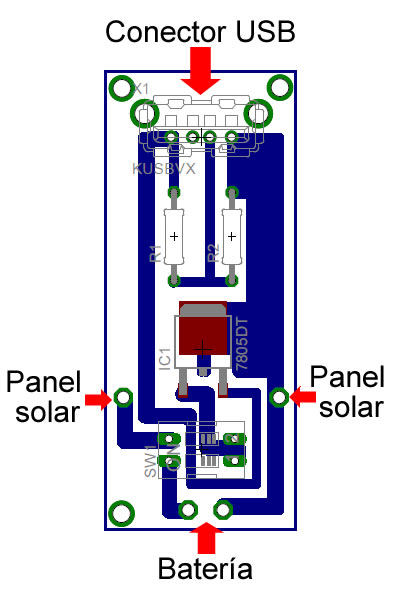
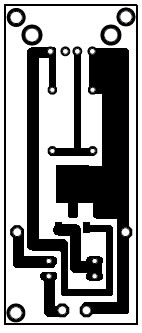
 

Figura 24: Diseño del circuito y de la PCB del cargador en Eagle Professional.

# CAPÍTULO 4. DESCRIPCIÓN DETALLADA DE LAS ACTIVIDADES

1. **Investigar los componentes a ocupar.**

Se realizó una investigación acerca de todos los componentes necesarios para desarrollar el proyecto, desde circuitos hasta dispositivos que se requieran para la consumación del prototipo, buscando siempre por aquellos componentes que resulten más económicos y más accesibles, así como también observar la elaboración de cada uno de los circuitos que se necesitarán.

1. **Realizar presupuesto.**

Cuando ya se tuvo la información de los componentes, se procedió a buscar los distintos distribuidores de material para decidir cuáles son los que se adaptan mejor a las necesidades del proyecto y de los desarrolladores.

1. **Obtener el material a utilizar.**

Una vez identificado cada uno de los distribuidores se adquirió el material, ya sea por paquetería u obteniéndolo directamente de las tiendas proveedoras, por lo que se tomó más tiempo del que se tenía planeado.

1. **Construir el circuito GPS con un Intel Galileo.**

Con los componentes ya listos, se construyó un módulo de GPS que se comunica con la computadora por medio de un Intel Galileo el cual tiene las facilidades de un microprocesador Arduino pero el poder de un microprocesador Intel, facilitando así la velocidad de comunicación entre los componentes sin utilizar demasiada programación o espacio.

1. **Construir el circuito de almacenamiento de energía con paneles solares.**

Se armó un circuito que obtenga la energía de unos paneles solares y la almacene para que esta pueda ser utilizada posteriormente para cargar cualquier dispositivo móvil como un celular, un mp3, un iPhone o inclusive una Tablet.

1. **Adaptar una mochila a una bicicleta plegable.**

Se requirió el encontrar una bicicleta que se adapte fácilmente a una mochila o viceversa, además de analizar si es menos complicado el obtener una bicicleta y una mochila y adaptarlas o si ya adquirir la bicicleta plegable con mochila incluida.

1. **Combinar todos los circuitos para el prototipo.**

Al tener los circuitos separados se ideó la forma en la que todos estén en un mismo lugar para que ocupen el mínimo de espacio y para que sea sencillo su almacenamiento y acoplamiento a la mochila.

1. **Implementar el prototipo desarrollado.**

Se procedió a realizar mejoras al proyecto como disminución de componentes o de energía a utilizar o cualquier mejora que se pueda efectuar para que el proyecto quede listo para su presentación.

1. **Documentar cada una de las fases.**

Cada que se realizó algo dentro del proyecto se documentó en tiempo y en forma para tener listo el protocolo de investigación del mismo para presentarlo al momento de culminar el proyecto.

# CAPÍTULO 5. RESULTADOS Y CONCLUSIONES

## 5.1 RESULTADOS

Se logró realizar el prototipo de lo que consistía en la elaboración de un cargador solar, el cual al contar con un puerto USB es capaz de cargar la batería de dispositivos móviles tales como teléfonos celulares y reproductores Mp3 de 5 volts. Estos módulos comprenden a los colectores solares, que son los paneles fotovoltaicos usados para generar energía, estos están compuestos por numerosas celdas que transforman la luz en electricidad .

El proyecto es un dispositivo que se llevó a cabo con un gasto económico mínimo, además de haber implementado el geolocalizador (GPS) exitosamente, por lo cual este diseño fue práctico para colocarse no solamente en una bicicleta para excursionistas como se planteó en un inicio, sino también en las paradas del autobús o en los patios de las escuelas, siendo inclusive un dispositivo importante para aquellas personas que puedan presentar una emergencia y necesiten hacer una llamada el lugares que no cuenten con el servicio de corriente eléctrica.

El proyecto no alcanzó su etapa final debido a la falta de presupuesto, por lo que no se consiguió la bicicleta plegable para así adaptarla al módulo GPS y así tener el prototipo final. Solamente se desarrollaron la etapa del geolocalizador en tiempo real y la etapa del cargador de dispositivos con paneles solares.

## 5.2 CONCLUSIONES

La sociedad actual vive en un entorno en el que se ha dado cuenta que las energías renovables son el mejor camino al futuro, debido a que son energías más limpias, debido a que no involucran un daño ambiental, ente otras cosas. Una excelente forma de utilizar la energía solar es mediante cargadores solares, ya que estos pueden alimentar dispositivos móviles de bajo consumo con facilidad, a la vez que alimentan en tiempo real una batería de emergencia y otros accesorios que puedan utilizarse en lugares donde no se encuentren enchufes eléctricos, como por ejemplo un dispositivo GPS. Sin duda alguna los resultados que se dieron en esta investigación fueron exitosos y con ello se cumple parcialmente el objetivo del trabajo.

# REFERENCIAS

* David Houlding (2013), GPS location sensing with ITEAD GPS Shield and Arduino Uno, Internet of Things, Recuperado el 22 de Febrero de 2015, de: http://davidhoulding.blogspot.mx/2014/06/gps-location-sensing-with-itead-gps.html
* Miguel Auza (2012), Arduino GPS Tutorial, All About EE, Recuperado el 23 de Febrero de 2015, de: http://allaboutee.com/2012/12/03/arduino-gps-tutorial-get-latitude-and-longitude-coordinates/
* Adilson Felipe (2015), Usando un Arduino GPS Shield con Google Earth, FilipeFlop, Recuperado el 23 de Febrero de 2015, de: http://blog.filipeflop.com/arduino/usando-o-arduino-gps-shield-com-google-earth.html
* Biospin (2014), Cargador Solar USB en menos de 20 minutos, Biospin, Recuperado el 25 de Marzo de 2015, de: http://www.bioespin.com/cargador-solar-usb-en-20-minutos.html

# ANEXOS

## Código a utilizar para el GPS sin pantalla.

#include <string.h>

#include <ctype.h>

int ledPin = 13; // LED test pin

int rxPin = 0; // RX PIN

int txPin = 1; // TX TX

int byteGPS=-1;

char linea[300] = "";

char comandoGPR[7] = "$GPRMC";

int cont=0;

int bien=0;

int conta=0;

int indices[13];

void setup() {

pinMode(ledPin, OUTPUT); // Initialize LED pin

pinMode(rxPin, INPUT);

pinMode(txPin, OUTPUT);

Serial.begin(4800);

for (int i=0;i<300;i++){ // Initialize a buffer for received data

linea[i]=' ';

}

}

void loop() {

digitalWrite(ledPin, HIGH);

byteGPS=Serial.read(); // Read a byte of the serial port

if (byteGPS == -1) { // See if the port is empty yet

delay(100);

} else {

// note: there is a potential buffer overflow here!

linea[conta]=byteGPS; // If there is serial port data, it is put in the buffer

conta++;

Serial.write(byteGPS);

if (byteGPS==13){ // If the received byte is = to 13, end of transmission

// note: the actual end of transmission is <CR><LF> (i.e. 0x13 0x10)

digitalWrite(ledPin, LOW);

cont=0;

bien=0;

// The following for loop starts at 1, because this code is clowny and the first byte is the <LF> (0x10) from the previous transmission.

for (int i=1;i<7;i++){ // Verifies if the received command starts with $GPR

if (linea[i]==comandoGPR[i-1]){

bien++;

}

}

if(bien==6){ // If yes, continue and process the data

for (int i=0;i<300;i++){

if (linea[i]==','){ // check for the position of the "," separator

// note: again, there is a potential buffer overflow here!

indices[cont]=i;

cont++;

}

if (linea[i]=='\*'){ // ... and the "\*"

indices[12]=i;

cont++;

}

}

Serial.println(""); // ... and write to the serial port

Serial.println("");

Serial.println("---------------");

for (int i=0;i<12;i++){

switch(i){

case 0 :Serial.print("Time in UTC (HhMmSs): ");break;

case 1 :Serial.print("Status (A=OK,V=KO): ");break;

case 2 :Serial.print("Latitude: ");break;

case 3 :Serial.print("Direction (N/S): ");break;

case 4 :Serial.print("Longitude: ");break;

case 5 :Serial.print("Direction (E/W): ");break;

case 6 :Serial.print("Velocity in knots: ");break;

case 7 :Serial.print("Heading in degrees: ");break;

case 8 :Serial.print("Date UTC (DdMmAa): ");break;

case 9 :Serial.print("Magnetic degrees: ");break;

case 10 :Serial.print("(E/W): ");break;

case 11 :Serial.print("Mode: ");break;

case 12 :Serial.print("Checksum: ");break;

}

for (int j=indices[i];j<(indices[i+1]-1);j++){

Serial.print(linea[j+1]);

}

Serial.println("");

}

Serial.println("---------------");

}

conta=0; // Reset the buffer

for (int i=0;i<300;i++){ //

linea[i]=' ';

}

}

}

}

## Código a utilizar para el GPS con la pantalla Display.

#include <SoftwareSerial.h>

#include <LiquidCrystal.h>

#include "TinyGPS.h"

TinyGPS gps;

int unoRxPin = 6; // conectar al Pin Tx del GPS

int unoTxPin = 7; // conectar al Pin Rx del GPS

SoftwareSerial ss(unoRxPin, unoTxPin);

LiquidCrystal lcd(12, 11, 10, 5, 4, 3, 2);

int backLight = 13; // Pin 13 controla el backlight

long startMillis;

long secondsToFirstLocation = 0;

void setup()

{

ss.begin(9600);

pinMode(backLight, OUTPUT);

digitalWrite(backLight, HIGH

lcd.begin(20,4); // columns, rows. use 16,2 for a 16x2 LCD, etc.

lcd.clear(); // Inicia con la pantalla negra

startMillis = millis();

}

void loop()

{

bool newData = false;

unsigned long chars = 0;

unsigned short sentences, failed;

// For one second we parse GPS data and report some key values

for (unsigned long start = millis(); millis() - start < 1000;)

{

while (ss.available())

{

int c = ss.read();

++chars;

if (gps.encode(c))

newData = true;

}

}

if (newData)

{

// we have a location fix so output the lat / long and time to acquire

if(secondsToFirstLocation == 0){

secondsToFirstLocation = (millis() - startMillis) / 1000;

}

lcd.clear(); // start with a blank screen

float flat, flon;

unsigned long age;

gps.f\_get\_position(&flat, &flon, &age);

lcd.setCursor(0,0);

lcd.print("Lat=");

lcd.print(flat == TinyGPS::GPS\_INVALID\_F\_ANGLE ? 0.0 : flat, 6);

lcd.setCursor(0,1);

lcd.print("Long=");

lcd.print(flon == TinyGPS::GPS\_INVALID\_F\_ANGLE ? 0.0 : flon, 6);

lcd.setCursor(0,2);

lcd.print("Acquire Time=");

lcd.print(secondsToFirstLocation);

lcd.print("s");

}

if (chars == 0){

// if you haven't got any chars then likely a wiring issue

lcd.setCursor(0,0);

lcd.print("No GPS: check wiring");

}

else if(secondsToFirstLocation == 0){

// if you have received some chars but not yet got a fix then indicate still searching and elapsed time

lcd.clear(); // start with a blank screen

long seconds = (millis() - startMillis) / 1000;

lcd.setCursor(0,0);

lcd.print("Searching ");

for(int i = 0; i < seconds % 4; ++i){

lcd.print(".");

}

lcd.setCursor(0,1);

lcd.print("Elapsed time:");

lcd.print(seconds);

lcd.print("s");

}

}