



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

“DESARROLLO DE SISTEMA GPS PARA APLICACIONES DE MONITOREO”

TESIS

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN TELECOMUNICACIONES

PRESENTA:

“ROBERTO ADRIAN ORTEGA LAZOS”

DIRECTOR DE TESIS

PH. D. CARLOS ROMO FUENTES

CIUDAD UNIVERSITARIA 30 /11/2014.



Tabla de contenido

Tabla de Figuras	iv
Agradecimientos	viii
Introducción	1
Justificación y Alcances del Proyecto.....	1
Descripción General del Proyecto	2
Objetivos del Proyecto.....	3
Capítulo 1 Sistemas de Posicionamiento por Satélite	4
El Sistema de Posicionamiento Global por Satélite	4
Constitución de un GNSS.....	4
Principio de Operación de un sistema GNSS.....	5
Antecedentes Históricos	7
Versiones de GNSS	8
Global Positioning System (GPS).....	10
Segmento espacial.....	10
Segmento de control.....	12
Segmento del usuario.....	14
El receptor GPS	15
Elementos esenciales de un receptor GPS.....	15
Clasificación de los receptores GPS.....	18
Especificaciones técnicas de los receptores GPS utilizados	18
Protocolo de comunicaciones NMEA 0183.....	27
El estándar NMEA 0183.....	27
Interfaz eléctrica.....	28
Formato general de los mensajes	29
Formato del mensaje GPS	29

Capítulo 2 El Microprocesador ATmega328	31
El microprocesador	31
Características técnicas del microprocesador ATmega328	32
Diagrama a bloques del microcontrolador ATmega328	33
Asignación de pines para configuración de 28 pines	34
Funciones de los puertos del microcontrolador ATmega328	35
Relación de pines ATmega328 con Arduino.....	37
Capítulo 3 Sistemas receptores GPS.....	38
Sistema con módulo receptor EM-408	38
Sistema con módulo receptor EM-408, arduino MEGA y compatibilidad micro SD.....	42
Sistema con Shield iTeed Studio y antena externa.....	45
Capítulo 4 Integración del Sistema Simplificado	47
Microprocesador	48
Receptor GPS	53
Interfaz micro SD	55
Energización del sistema.....	61
Diagrama del sistema receptor GPS	62
Diseño de PCB.....	63
Esquemáticos diseñados en Altium Designer 14	64
Archivos “GERBER” para producción de PCB	71
Capítulo 5 Pruebas y Resultados	76
Protocolo de Validación.....	76
Receptor EM-408	80
Receptor EM-408 con interfaz micro SD y arduino Mega	83
Receptor iTeed Studio.....	89
Prototipo de Módulo GPS.....	93
Conclusiones	100

Anexos.....	103
Acrónimos	118
Bibliografía	119

Tabla de Figuras

Capítulo 1 Sistemas de Posicionamiento por Satélite

Figura 1.1 Elementos de un sistema de navegación por satélite.....	5
Figura 1.2 Mediciones que se pueden realizar entre un satélite y un receptor.....	5
Figura 1.3 A) Técnica de medición utilizando un satélite B) Técnica de medición utilizando varios satélites.....	6
Figura 1.4 A) Se desea determinar la órbita del satélite Sputnik B) Conocida la órbita del satélite se desea determinar la posición de un objeto.....	7
Figura 1.5 Constelación de satélites GPS.....	10
Figura 1.6 Distribución del segmento de control.....	13
Figura 1.7 Diagrama a bloques de los componentes principales de un receptor GPS	15
Figura 1.8 Dimensiones físicas del receptor EM-408.....	20
Figura 1.9 A) Conector hembra B) Conector macho.....	20
Figura 1.10 A) Receptor EM-408 B) Mapa de pines del receptor EM-408.....	21
Figura 1.11 ITead Studio GPS shield, Arduino UNO y antena externa.....	21
Figura 1.12 Footprint de la tarjeta iTead studio.....	23
Figura 1.13 Receptor EM-506.....	24
Figura 1.14 Dimensiones físicas del receptor EM-506.....	26
Figura 1.15 A) Conector macho B) Conector hembra.....	26
Figura 1.16 Asignación de pines del receptor EM-506.....	27

Capítulo 2 El microprocesador ATmega328

Figura 2.1 Diagrama a bloques del microcontrolador ATmega328.....	33
Figura 2.2 Mapa de pines del microcontrolador ATmega328 versión de empaquetado 28 pines.....	34
Figura 2.3 Relación de pines de microcontrolador ATmega328 con pines Arduino UNO.....	37

Capítulo 3 Sistemas Receptores GPS

Figura 3.1 Piezas incluidas en el kit del shield GPS de sparkfun.....	38
Figura 3.2 Shield GPS sparkfun con sus respectivos headers soldados de manera correcta.....	39
Figura 3.3 Conector vertical SMD compatible con receptor EM-408.....	39
Figura 3.4 Shield GPS de sparkfun conectado a un módulo receptor EM-408..	40
Figura 3.5 Módulo receptor EM-408 ensamblado con shield GPS sparkfun y Arduino UNO.....	40
Figura 3.6 Wireless SD shield de Arduino.....	42
Figura 3.7 Arduino Mega.....	43

Figura 3.8 Arduino Mega conectado a Wireless SD shield y módulo receptor EM-408.....	44
Figura 3.9 Antena GPS 3 [V] con conector SMA.....	45
Figura 3.10 Shield ITead studio con antena GPS externa conectado a Arduino UNO.....	46

Capítulo 4 Integración de Sistema Simplificado

Figura 4.1 Esquemático de tarjeta Arduino UNO R3.....	48
Figura 4.2 Mapa de pines de microcontrolador ATmega328 versión de empaquetado 28 pines.....	49
Figura 4.3 Relación de pines de microcontrolador ATmega328 con pines Arduino UNO.....	50
Figura 4.4 Conexión mínima de microcontrolador ATmega328 conectado a Arduino UNO.....	51
Figura 4.5 Conexiones necesarias para grabar el gestor de arranque del microcontrolador ATmega328.....	52
Figura 4.6 Receptor EM-506.....	54
Figura 4.7 Asignación de pines de una tarjeta micro SD.....	55
Figura 4.8 Esquemático de interfaz micro SD del iTeard Studio GPS shield.....	56
Figura 4.9 Esquemático de seleccionador de voltaje del iTeard Studio GPS shield.....	57
Figura 4.10 Esquemático de conexión entre interfaz micro SD y header de arduino uno del iTeard Studio GPS shield.....	57
Figura 4.11 Esquemático de conexiones con tarjeta micro SD en Wireless shield.....	58
Figura 4.12 A) Conexiones de micro SD con arduino en Wireless shield B) Conexiones de energización de Wireless shield.....	58
Figura 4.13 Regulador de voltaje a 3.3 [V] en Wireless shield.....	59
Figura 4.14 Asignación de pines de circuito integrado SNLVC125A A) Empaquetado D, DB, NS o PW B) Empaquetado RGY.....	59
Figura 4.15 Diagrama lógico de cada buffer de CI SNLVC125A.....	60
Figura 4.16 Diagrama a bloques del sistema GPS con interfaz micro SD.....	62
Figura 4.17 Diagrama a bloques del sistema simplificado GPS conectado al sistema de monitoreo.....	63
Figura 4.18 Logo Altium Designer 14.....	63
Figura 4.19 Diseño propuesto de PCB, módulo receptor GPS.....	71
Figura 4.20 Archivo GERBER de sujetador de tarjeta micro SD.....	72
Figura 4.21 Archivo camtastic .drill.....	73
Figura 4.22 Archivo camtastic .gtl.....	73
Figura 4.23 Archivo camtastic .gto.....	74
Figura 4.24 Archivo camtastic .gts.....	74

Figura 4.25 Archivo camtastic .cam.....	75
---	----

Capítulo 5 Pruebas y Resultados

Figura 5.1 Obtención de datos de receptor BU353 con GPSinfo.....	77
Figura 5.2 Receptor BU353 de Globalsat.....	78
Figura 5.3 Página de inicio de Google Maps México.....	79
Figura 5.4 Página de inicio de Pixelis Maps.....	79
Figura 5.5 Monitor serial con datos recibidos por EM-408, buen clima y línea de vista.....	80
Figura 5.6 Monitor serial con datos recibidos por EM-408, lluvia densa y línea de vista.....	80
Figura 5.7 Monitor serial con todas las cadenas de datos recibidas por EM-408.....	81
Figura 5.8 Mapa generado con Pixelis Maps.....	82
Figura 5.9 Mapa generado con Google Maps.....	83
Figura 5.10 Monitor serial con datos recibidos por Arduino Mega y receptor EM-408.....	84
Figura 5.11 Monitor serial con datos recibidos por Arduino Mega y receptor EM-408 sin tarjeta SD.....	85
Figura 5.12 Monitor serial con datos recibidos por Arduino Mega tarjeta SD y receptor desconectado.....	85
Figura 5.13 Archivo LOG.csv generado por sistema receptor con EM-408 y Arduino Mega.....	86
Figura 5.14 Archivo LOG.csv generado por sistema receptor con EM-408 y Arduino Mega con encabezados.....	87
Figura 5.15 Pantalla de inicio GPS visualizer.....	87
Figura 5.16 Mapa extensión KML generado con GPS visualizer.....	88
Figura 5.17 Mapa en Google Maps generado con GPS visualizer.....	89
Figura 5.18 Monitor serial con datos recibidos por iTeed shield.....	90
Figura 5.19 Mapa generado con Pixelis Maps a partir de datos recabados con receptor iTeed Studio.....	90
Figura 5.20 Posición generada con receptor BU353 interpretada con GPSinfo...91	
Figura 5.21 Monitor serial con todas las cadenas de datos recibidas por iTeed Shield.....	92
Figura 5.22 Diseño propuesto de PCB, módulo receptor GPS.....	93
Figura 5.23 Prototipo de módulo GPS.....	94
Figura 5.24 Arduino UNO sin microcontrolador ATmega328.....	95
Figura 5.25 Diagrama de conexiones requeridas para comunicar prototipo GPS con computadora mediante tarjeta Arduino UNO.....	95
Figura 5.26 Prototipo de módulo GPS conectado a receptor iTeed Studio con antena externa y Arduino UNO.....	96

Figura 5.27 Datos de posición obtenidos mediante prototipo del módulo receptor GPS.....	97
Figura 5.28 Mapa generado por Pixelis Maps con la posición obtenida con el prototipo del módulo GPS.....	98
Figura 5.29 Posición y mapa obtenidos con Google Maps.....	98
Figura 5.30 Monitor serial con datos recibidos situados en interior con prototipo de módulo GPS.....	99

Agradecimientos

El autor desea expresar su más sincero agradecimiento a

Introducción

Justificación y Alcances del Proyecto

Actualmente existe una tendencia a desarrollar sistemas aplicados a monitoreo para optimizar procesos, prever fallas o recurrir a realizar ajustes al sistema que se esté monitoreando; tal es el caso de sistemas de telemetría con sus infinitas posibilidades de aplicación, desde sistemas de telemetría satelital hasta sistemas de telemetría para redes ad-hoc.

Para estos sistemas resulta fundamental conocer la ubicación del elemento del sistema del cual se está recabando la información y es aquí donde esta tesis adquiere justificación.

Sería de gran utilidad otorgar la capacidad de geo localización al sistema de monitoreo en cuestión ya que al indicarnos donde se están realizando las lecturas, discriminando entre múltiples elementos de censado del sistema de monitoreo, no haría falta ir a una base de datos para saber dónde se encuentra ese elemento por lo que se podría dar algún ajuste o mantenimiento al sistema que se esté monitoreando y por lo tanto se podría automatizar el sistema de monitoreo reduciendo la participación de recursos humanos y el uso del capital que lo anterior representa.

El conocer la ubicación de los elementos de un sistema de monitoreo también nos permitiría, de ser necesario, canalizar de una manera más eficiente los recursos humanos necesarios para dar mantenimiento a los elementos del sistema monitoreado ya que se podrá distinguir de donde provienen los datos recabados por el sistema de monitoreo. No será muy útil saber que se tiene que dar mantenimiento a algún elemento del sistema monitoreado si no se conoce donde se encuentra la falla.

Por lo que para conocer la posición de cualquier elemento del sistema que se está monitoreando se debe equipar al sistema de monitoreo con un sistema de localización el cual brinde la información deseada como es latitud, longitud, altitud, velocidad, tiempo del satélite, hora local entre otros parámetros.

Se requiere un sistema de localización que se pueda integrar fácilmente al sistema de monitoreo, que tenga dimensiones reducidas, que sea programable para satisfacer las diferentes necesidades generadas por los proyectos y que además sea un sistema económico.

Descripción General del Proyecto

Para ofrecer una solución a la problemática planteada primeramente se realizó una extensa investigación sobre el funcionamiento de los sistemas de navegación global por satélite (GNSS) como el sistema de posicionamiento global (GPS) y otras versiones como el GALILEO y GLONASS.

A partir de esa información se optó por trabajar con el sistema de posicionamiento global GPS debido a su capacidad de cobertura precisión y otros parámetros que resultaron puntos clave a lo largo del desarrollo del proyecto como es el costo de los módulos receptores y la compatibilidad de estos con la tecnología arduino.

Contando con los conocimientos relacionados a sistemas GNSS se trabajó con sistemas receptores amigables con tecnología arduino ya que esta plataforma además de ser software libre, es una placa reprogramable con la que se pudo experimentar y hacer diferentes pruebas con las que se obtuvo un entendimiento mayor de estos sistemas.

Se programaron las tarjetas de desarrollo arduino en sus versiones UNO y MEGA para hacerlas funcionar junto con diferentes tipos de receptores así como diferentes tipos de antenas lo que permitió evaluar las ventajas y desventajas que ofrecen en cada una de sus versiones esto para poder ofrecer un sistema óptimo que resuelva la problemática planteada en este proyecto.

Con base a los resultados anteriores se seleccionó uno de los sistemas con los cuales se trabajó, para simplificarlo y que diera solución a la problemática del proyecto. Se desarrolló e implemento un prototipo para realizar pruebas de funcionamiento y por último se propuso un diseño de circuito impreso el cual se podrá utilizar para integrarse a un sistema de monitoreo.

Lo anterior se puede ver en cuatro grandes etapas del proyecto:

- 1) Investigación y documentación de sistemas GNSS, receptores GPS y tecnología arduino
- 2) Experimentación y pruebas con módulos receptores GPS, diferentes tipos de antenas así como con las versiones UNO y MEGA de la tarjeta de desarrollo arduino
- 3) Selección de equipo para diseñar y realizar un sistema simplificado
- 4) Diseño de circuito impreso

Objetivos del Proyecto

Principalmente se fijaron dos objetivos a cumplir en este proyecto:

- 1) El Desarrollo e implementación de un módulo receptor con base en el sistema de posicionamiento global GPS compatible con tecnología arduino para otorgarle a un sistema de monitoreo de parámetros físicos la funcionalidad de geo localización.
- 2) Realizar un documento, el cual sirva como referencia para proyectos de ingeniería relacionados con sistemas receptores GPS.

Se espera que el modulo receptor sea confiable, con las menores dimensiones posibles para que su adaptación al sistema de monitoreo sea más sencilla, que sea un diseño sencillo y que a su vez sea una propuesta económica para solucionar la problemática planteada.

Este sistema deberá proporcionar la información que el sistema de monitoreo requiera como puede ser latitud, longitud, altitud, velocidad, tiempo del satélite, hora local entre otros.

Como no se cuenta exactamente con las necesidades del sistema de monitoreo al cual se integrara nuestro sistema se deberá ofrecer un sistema con la capacidad de ser reprogramado fácilmente y que se pueda adaptar a las necesidades de los proyectos en los que se quiera implementar.

Capítulo 1 Sistemas de Posicionamiento por Satélite

El Sistema de Posicionamiento Global por Satélite

El sistema GNSS denominado así por sus siglas en inglés (Global Navigation Satellite System) es una constelación de satélites los cuales transmiten señales con las que es posible determinar la posición de un objeto en cualquier parte del globo terrestre ya sea en mar aire o tierra.

Este sistema permite determinar las coordenadas geográficas y la altitud de un punto gracias a la recepción de señales provenientes de constelaciones de satélites artificiales de la Tierra para fines de radio navegación, transporte, geodésicos, hidrográficos, agrícolas, y otras actividades.

Un sistema de navegación basado en satélites artificiales puede proporcionar a los usuarios información sobre la posición y la hora con una gran exactitud, en cualquier parte del mundo, las 24 horas del día y en todas las condiciones climatológicas aunque el sistema se verá afectado por algunas condiciones como por ejemplo, si se tiene línea de vista directa con los satélites o no, cantidad de satélites disponibles o el entorno donde se encuentre el receptor.

Constitución de un GNSS

Un sistema de navegación por satélite está constituido principalmente por:

- **Segmento Espacial:** Satélites
Radiodifusión de las señales
- **Segmento de Control:** Estaciones de seguimiento
Seguimiento y control de los satélites
Transmisión de la información que se radiodifunde (Banda S y modulación de fase)
Cálculo de los parámetros orbitales (efemérides)
- **Segmento de Usuarios:** Equipos Embarcados
Receptor para determinar los parámetros eléctricos
Calculador para corregir errores y determinar la posición

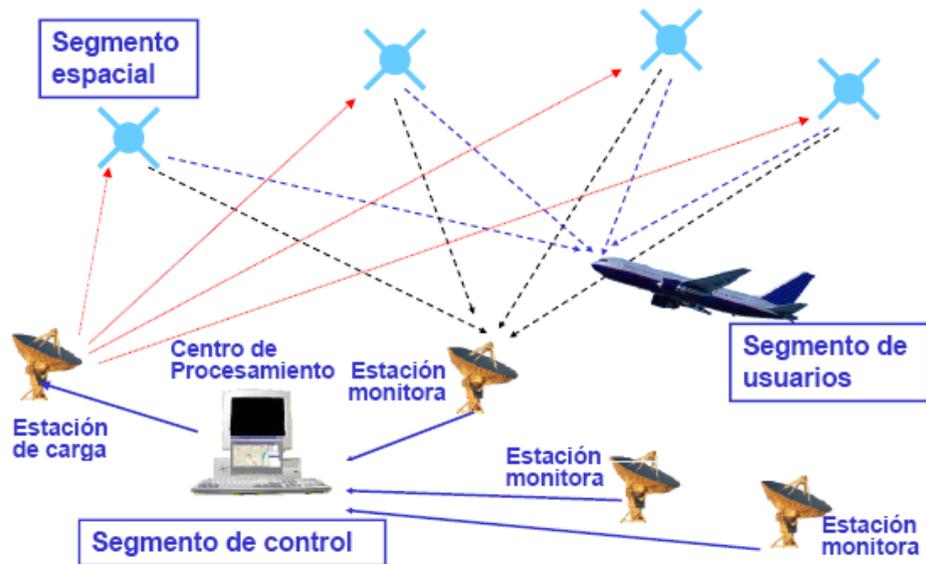
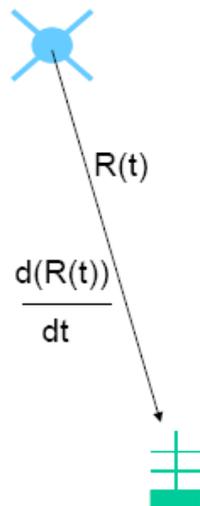


FIGURA 1.1 ELEMENTOS DE UN SISTEMA DE NAVEGACIÓN POR SATÉLITE

Principio de Operación de un sistema GNSS

Primero pensamos en las medidas que se pueden realizar:

- **Distancia satélite-receptor**, a través de los retardos temporales y/o desfases que sufre la señal durante su propagación.



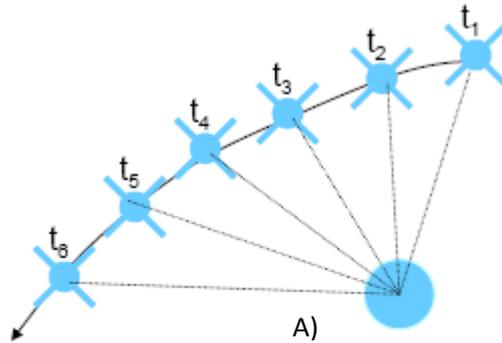
- **Velocidad de variación de esta distancia**, a través del desplazamiento doppler.

Entonces para calcular la posición:

- Cada medida permite obtener una superficie en la que está situado el receptor.
- La intersección de varias superficies permite obtener la posición.
- Las superficies deben cortarse adecuadamente por lo que normalmente se requiere 4 o más satélites.

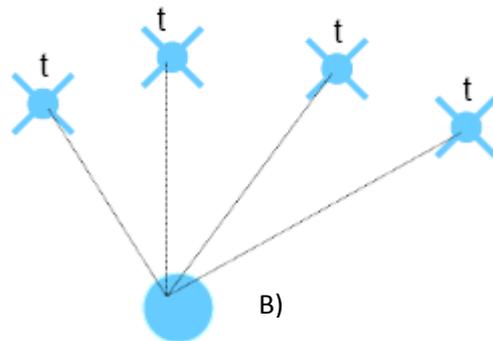
FIGURA 1.2 VALORES QUE SE PUEDEN REALIZAR ENTRE UN SATÉLITE Y UN RECEPTOR

Se tienen dos filosofías para realizar las mediciones anteriores:



Un solo satélite visible y medidas en diferentes instantes de tiempo para generar varias superficies de intersección.

Ej. TRANSIT 6 satélites con órbitas polares a 1100 Km de altura



Varios Satélites visibles simultáneamente. Necesita muchos satélites y genera medidas casi instantáneas y continuas.

Ej. GPS 18 a 24 satélites con órbitas a 20000 Km de altura.

FIGURA 1.3 A) TÉCNICA DE MEDICIÓN UTILIZANDO UN SATÉLITE
B) TÉCNICA DE MEDICIÓN UTILIZANDO VARIOS SATÉLITES

Características y aplicaciones de cada técnica:

- Desviación Doppler de una portadora emitida por una emisora móvil

Características	Aplicaciones
Simple	Posicionamiento inicial
Poco preciso	Navegación de baja dinámica
No instantáneo	

- Retardo de la señal

Características	Aplicaciones
Complejo	Navegación
Preciso	Posicionamiento preciso
Casi instantáneo	

- Desfasaje de una portadora

Características	Aplicaciones
Muy complejo	Topografía
Muy ambiguo	Geodesia
No instantáneo	
Medidas diferenciales	

Antecedentes Históricos

El primer sistema de Navegación por Satélite fue concebido durante los experimentos realizados en la Universidad Johns Hopkins para determinar la órbita del primer satélite artificial Sputnik-1, midiendo el desplazamiento Doppler de la señal de 20 [MHz] que emitía. Entonces se planteó invertir esa idea e intentar determinar la distancia de un objeto (receptor) conocida la órbita del primer satélite.

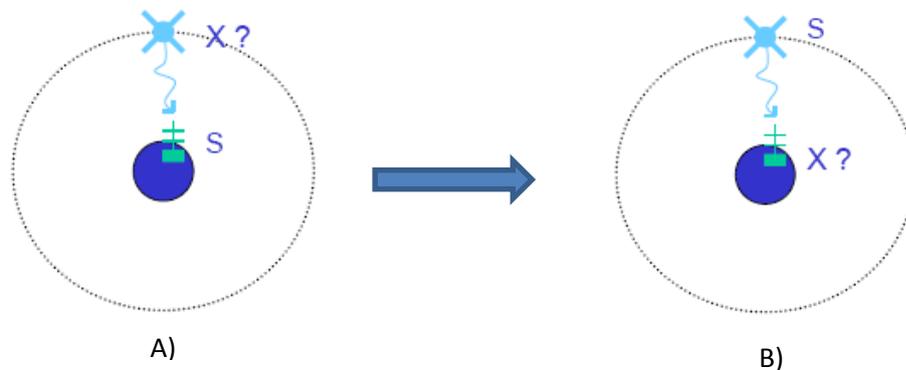


FIGURA 1.4 A) SE DESEA DETERMINAR LA ÓRBITA DEL SATÉLITE SPUTNIK

B) CONOCIDA LA ÓRBITA DEL SATÉLITE SE DESEA DETERMINAR LA POSICIÓN DE UN OBJETO

Ubicándonos en la guerra fría surge la necesidad por parte de los Estados Unidos de América del despliegue de los submarinos Polaris los cuales tenían la capacidad de realizar disparos nucleares sin tener que emerger del agua en respuesta a la posible amenaza de guerra nuclear con la Unión Soviética. Este tiempo fue sumamente tenso y complejo ya que para realizar algún ataque, el que decidiera realizarlo debía asegurar la completa destrucción del enemigo.

En consecuencia a lo anterior se desarrolla el sistema TRANSIT, utilizado por la marina de los EE.UU para conseguir información precisa para el lanzamiento de los misiles Polaris y para la navegación de los barcos y submarinos, aunque a partir de 1967 también se utilizó para estudios topográficos, geotécnicos e hidrográficos.

Desde los satélites del sistema TRANSIT se transmiten dos señales portadoras (UHF) periódicamente (cada dos minutos). Las incidencias de la órbita y correcciones del reloj se actualizan dos veces al día, con esta información se calcula la posición del satélite en a lo largo del tiempo. Al utilizar dos señales se reduce el número de errores. Este sistema, hizo posible la sincronización de los relojes en todo el mundo con una precisión de 50 microsegundos.

En 1958 se realizó el lanzamiento del primer satélite y para 1964 el sistema TRANSIT entra en operación con una flota de 10 satélites.

En 1973 entra el programa NAVSTAR-GPS que fue una serie de 24 satélites estadounidenses de navegación que completan el Sistema de posicionamiento global (Global Positioning System, GPS). Permiten conocer a los navegantes su posición en la Tierra con un error/margen de 10 m, la velocidad con un error hasta de 0,1 m/s, precisando el tiempo hasta la millonésima de segundo. Se encuentran a una altura de unos 20.200 km, y completan una órbita a la Tierra en 12 horas.

Durante el Día del Trabajo (1 de mayo) de 1973, se encontraron doce oficiales militares del Pentágono quienes discutieron la creación de un Sistema Satelital de Navegación de Defensa. Fue en esa reunión que la verdadera síntesis de creación del GPS fue llevada a cabo. Más tarde en ese año, el programa DNSS se nombró NAVSTAR. Con los satélites individuales asociados con el nombre NAVSTAR, así como sus predecesores TRANSIT y Timation, se utilizó un nombre para identificar la constelación de satélites NAVSTAR: NAVSTAR-GPS, y más tarde se redujo simplemente a GPS.

Versiones de GNSS

- U.S.A.: **GPS** es un sistema perteneciente al gobierno de los estados unidos de américa y funciona mediante una red de 24 satélites en órbita sobre el planeta tierra, a 20.200 km, con trayectorias sincronizadas para cubrir toda la superficie de la Tierra.
- Unión Europea: **Galileo** es un sistema global de navegación por satélite (GNSS) desarrollado por la Unión Europea (UE), con el objeto de evitar la dependencia de los sistemas GPS y GLONASS.1 Al contrario de estos dos, será de uso civil. El sistema se espera poner en marcha en 2014

después de sufrir una serie de reveses técnicos y políticos para su puesta en marcha.

- Rusia: **GLONASS** (acrónimo en ruso: ГЛОНАСС, ГЛОбальная НАвигационная Спутниковая Система tr.: Global'naya Navigatsionnaya Sputnikovaya Sistema) es un Sistema Global de Navegación por Satélite desarrollado por la Unión Soviética, siendo hoy administrado por la Federación Rusa y que representa la contraparte del GPS estadounidense y del futuro Galileo europeo.

Consta de una constelación de 31 satélites: 24 en activo, 3 satélites de repuesto, 2 en mantenimiento, uno en servicio y uno en pruebas, situados en tres planos orbitales con 8 satélites cada uno y siguiendo una órbita inclinada de 64,8° con un radio de 25510 km.

La constelación de GLONASS se mueve en órbita alrededor de la Tierra con una altitud de 19100 km algo más bajo que el GPS (20.200 km) y tarda aproximadamente 11 horas y 15 minutos en completar una órbita. El sistema está a cargo del Ministerio de Defensa de la Federación Rusa y los satélites se han lanzado desde Baikonur, en Kazajistán.

- China: **Beidou** es un proyecto desarrollado por la República Popular de China para obtener un sistema de navegación por satélite. "Beidou" es el nombre chino para la constelación de la Osa Mayor. La primera generación, BeiDou-1, ya está operativa desde el 2000 y es un sistema de posicionamiento por satélite local dando servicio a China y a sus países vecinos. La segunda generación, también llamada Compass o BeiDou-2, será un sistema de posicionamiento global con un funcionamiento similar al GPS.

A diferencia de los sistemas GPS, GLONASS, y GALILEO, que utilizan satélites en órbitas bajas y ofrecen servicio global, la primera generación, Beidou-1 usa satélites en órbita geoestacionaria. Esto implica que el sistema no requiera una gran constelación de satélites, pero limita su cobertura sobre la tierra a la visible por los satélites, China en este caso. Otra gran diferencia de BeiDou-1 es que calcula las coordenadas únicamente con dos satélites y una estación en tierra. Esto implica la necesidad de enviar una señal desde el dispositivo remoto, cosa que no es necesaria con GPS o GLONASS.

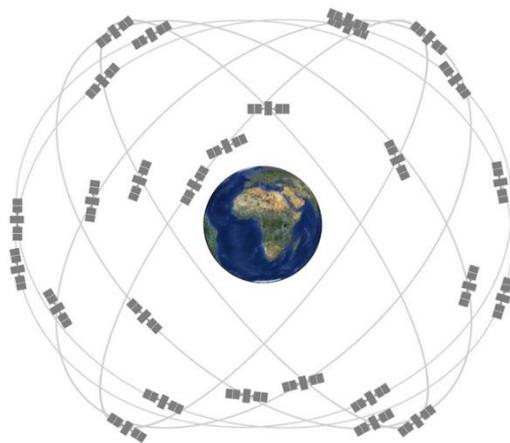
Global Positioning System (GPS)

El Sistema de Posicionamiento Global (GPS) es un servicio propiedad de los EE.UU. que proporciona a los usuarios información sobre posicionamiento, navegación y cronometría. Este sistema está constituido por tres segmentos: el segmento espacial, el segmento de control y el segmento del usuario. La Fuerza Aérea de los Estados Unidos desarrolla, mantiene y opera los segmentos espacial y de control.

Segmento espacial

El segmento espacial del sistema GPS consiste en una constelación de satélites transmitiendo señales de radio a usuarios. Los Estados Unidos de América están comprometidos para tener al menos 24 satélites GPS en operación al 95% de tiempo.

Los satélites de GPS operan en una órbita terrestre media (MEO) a una altitud aproximada de 20200 Km. Cada satélite circula la tierra dos veces al día.



Los satélites de la constelación GPS están distribuidos en seis planos equi-espaciados alrededor de la tierra.

Cada plano contiene 4 espacios ocupados por satélites. Este arreglo de 24 espacios asegura a los usuarios poder tener en línea de vista por lo menos 4 satélites desde cualquier punto en el planeta.

FIGURA 1.5 CONSTELACIÓN DE SATÉLITES GPS

Lo fuerza aérea normalmente opera más de 24 satélites para mantener una buena cobertura de servicio sin importar donde se encuentren los satélites de servicio. Los satélites extra pueden incrementar el desempeño del sistema GPS pero no son considerados parte del núcleo de la constelación.

En Junio del 2011 la Fuerza área realizo una expansión a la constelación GPS conocida como configuración “*Expandable 24*”. Tres de los 24 espacios fueron añadidos y seis satélites fueron reposicionados, por lo que tres de los satélites extra fueron añadidos a la constelación base. Como resultado, GPS actualmente opera con una constelación con 27 espacios mejorando así su cobertura en prácticamente todo el mundo.

SATÉLITES

La constelación GPS es una mezcla de satélites viejos y nuevos. En la tabla siguiente se muestran las diferentes generaciones de satélites y sus características generales serán mencionadas a continuación.

Satélites de Legado		Satélites Modernos	
Bloque IIA	Bloque IIR	Bloque IIR(M)	Bloque IIF
8 operacionales	12 operacionales	7 operacionales	4 operacionales

Bloque IIA

- Código de alta adquisición (C/A) en banda L1 para usuarios civiles.
- Código de precisión P(Y) en bandas L1 y L2 para usuarios militares.
- Diseño de tiempo de vida aprox. 7.5 años.
- Lanzados en 1990-1997

Bloque IIR

- Código de alta adquisición (C/A) en banda L1 para usuarios civiles.
- Código de precisión P(Y) en bandas L1 y L2 para usuarios militares.
- Monitoreo de reloj a bordo.
- Diseño de tiempo de vida aprox. 7.5 años.
- Lanzados en 1997-2004

Bloque IIR (M)

- Conserva todas las señales de los satélites de legado.
- Segunda señal civil en banda L2 (L2C)
- Nuevas señales en código militar M para mejorar resistencia de interferencia
- Niveles de potencia flexibles para señales militares
- Diseño de tiempo de vida aprox. 7.5 años.
- Lanzados en 2005-2009

Bloque IIF

- Todas las señales del bloque IIR (M)
- Tercera señal civil en banda L5
- Relojes atómicos avanzados
- Precisión, intensidad de señal y calidad mejoradas
- Diseño de tiempo de vida aprox. 12 años.
- Lanzados desde 2010

Actualmente se encuentra en producción un nuevo bloque llamado GPS III y contará con las siguientes características:

- Todas las señales del bloque IIF
- Cuarta señal civil en banda L1 (L1C)
- Sin disponibilidad selectiva
- Satélites 9+: laser reflectores; búsqueda y rescate de datos (DASS)
- Precisión, confiabilidad e integridad de señal mejoradas
- Diseño de tiempo de vida aprox. 15 años.
- Serán lanzados a partir de 2015

Segmento de control

El segmento de control del sistema GPS consiste en una red global de estaciones terrenas que rastrean a los satélites GPS, monitorean sus transmisiones, realizan análisis y mandan comandos así como datos a la constelación.

El segmento de control actual está constituido por:

Una estación maestra de control, una estación maestra de control alterna, 12 antenas de comando y control, y 16 sitios de monitoreo.

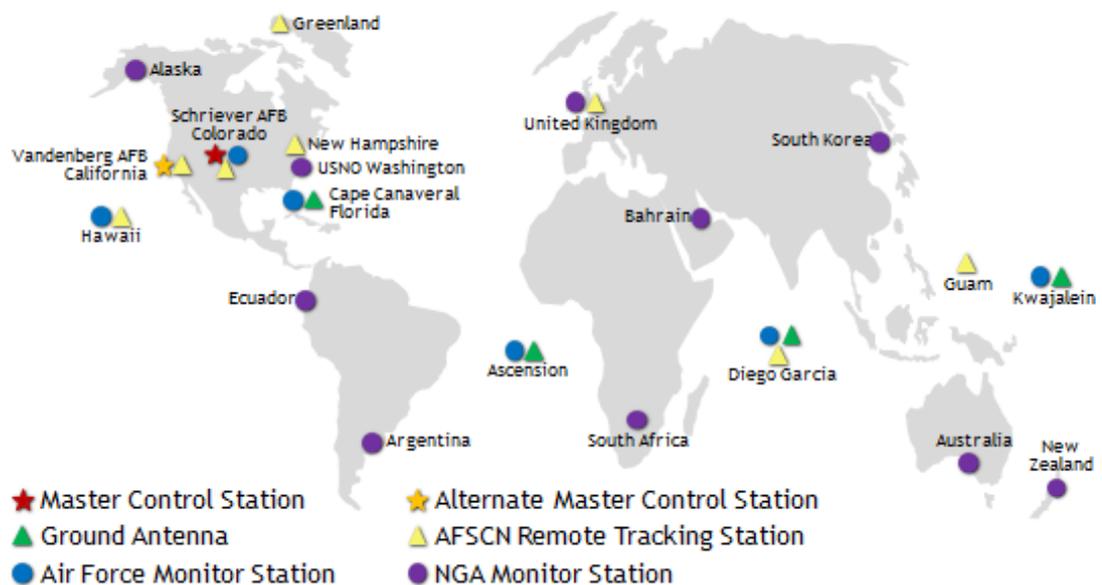


FIGURA 1.6 DISTRIBUCIÓN DEL SEGMENTO DE CONTROL

ELEMENTOS DEL SISTEMA DE CONTROL

ESTACION MAESTRA DE CONTROL (MCS)

La estación maestra de control en Colorado es donde el segundo escuadrón de operaciones espaciales (2SOPS) realiza las funciones de segmento de control primario, proporcionando comando y control de la constelación GPS.

Es aquí donde se evalúa el estado de salud de la constelación GPS, se realiza mantenimiento a los satélites y/o se resuelve cualquier anomalía.

La estación maestra de control tiene la función de reposicionar a los satélites para mantener en óptimo estado la constelación de satélites GPS.

ESTACIONES DE MONITOREO

Las estaciones de monitoreo siguen los satélites GPS conforme pasan y canalizan sus observaciones de vuelta a la estación maestra de control.

Las estaciones de monitoreo recolectan datos atmosféricos, medidas de portadora y distancia así como datos de navegación.

Hay 16 estaciones de monitoreo localizadas alrededor del mundo, incluyendo seis de la fuerza aérea y diez de la Agencia Nacional de Inteligencia Geoespacial.

ANTENAS TERRESTRES

Estas antenas son usadas para comunicación con los satélites GPS para propósitos de comando y control. Estas antenas utilizan enlaces en banda S con los que se envía o recibe datos de navegación y telemetría.

Hay 4 sitios dedicados a antenas terrestres GPS con una locación compartida con las estaciones de monitoreo en:

- Kwajalein Atoll
- Isla Ascensión
- Diego García
- Cabo Cañaveral

Además de lo anterior, el segmento de control está conectado a la red de control de satélite de la fuerza aérea (*AFSCN Air Force Satellite Control Network*) incrementando la visibilidad, flexibilidad y robustez para telemetría, rastreo y comandos.

Segmento del usuario

Este segmento se refiere a los distintos tipos de receptores que existen en el mercado y a los distintos usuarios del sistema GPS. Con el paso del tiempo se encuentran cada vez más aplicaciones del sistema, por lo que es necesario diseñar y desarrollar equipos con ciertas características para adaptarse a las nuevas y distintas necesidades de los usuarios.

Los satélites envían señales de radio a los receptores GPS para que estos puedan determinar qué tan lejos está cada satélite. Debido a que los satélites están orbitando a una distancia de 20,000 [Km] de la superficie de la tierra, las señales llegan débiles al receptor. Esto significa que el receptor debe hallarse en un área abierta para poder funcionar de manera correcta.

El receptor GPS

Elementos esenciales de un receptor GPS

Los principales componentes de un receptor GPS son los siguientes:

- Antena con preamplificador
- Sección de RF con identificación y procesamiento de señales
- Micro-procesador para: control del receptor, muestreo de datos y procesamiento de datos
- Oscilador de precisión
- Fuente de alimentación
- Interfaz de usuario, panel de comandos
- Dispositivo de memoria para almacenamiento de datos e instrucciones

Lo anterior se puede observar en el siguiente diagrama de bloques:

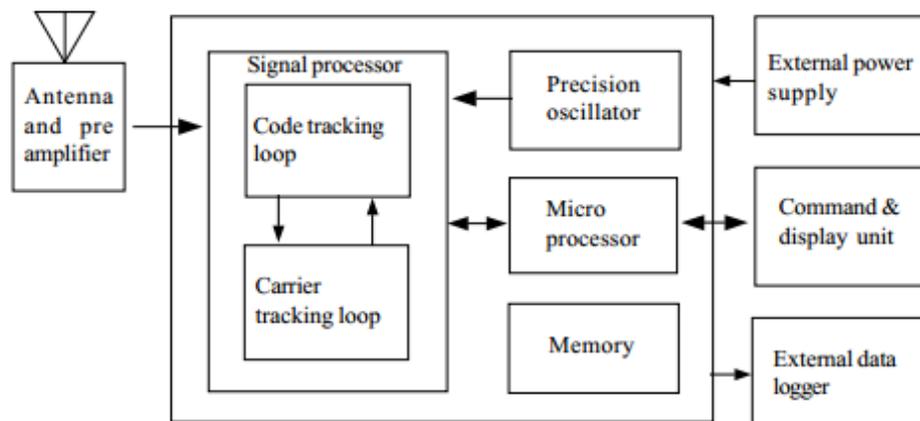


FIGURA 1.7 DIAGRAMA A BLOQUES COMPONENTES PRINCIPALES DE UN RECEPTOR GPS

Antena. La función de la antena es detectar las ondas electromagnéticas transmitidas por los satélites GPS y convertir la energía de la onda en corriente eléctrica. También tendrá que amplificar la intensidad de la señal y enviar la señal al siguiente bloque del receptor por lo que incluirá un preamplificador de bajo ruido con uno o más filtros asociados.

Existen diferentes tipos de antenas en los receptores GPS pero las principales son:

- Mono polo o dipolo
- Helicoidal
- Espiral
- Microcinta (de parche)
- *Choke rings*

De las antenas anteriores la más usada es la antena microcinta por sus ventajas de construcción, tamaño y materialización de receptor GPS.

Sección de RF con identificación y procesamiento de señales. Las señales GPS entrantes se convierten a una frecuencia inferior, llamada frecuencia intermedia ya que esta frecuencia es más fácil de manejar por los componentes del receptor.

El canal receptor es la unidad electrónica principal de un receptor GPS. Un receptor puede tener uno o más canales y existen tres modos de utilizar los canales:

- Receptor multicanal. En canal paralelo cada canal está siguiendo continuamente un satélite en particular y para que el receptor pueda determinar la posición y el tiempo se requiere un mínimo de cuatro canales paralelos (cuatro satélites en vista).
- Receptor secuencial. En canal secuencial, el canal cambia de satélite en intervalos regulares. Los receptores de un solo canal toman al menos cuatro veces de 30 segundos para establecer su posición, aunque algunos tipos de receptores tienen un canal designado para lectura de la señal de datos. En la mayoría de los casos, se utilizan canales de secuenciación rápida con una velocidad de conmutación de alrededor de un segundo por satélite.

- **Receptor Multiplexor.** En canal de multiplexado. Al multiplexar los canales un receptor conmuta secuencialmente sus canales disponibles a través de todas las señales de los satélites a una tasa muy rápida, de modo que los mensajes de todos los satélites pueden obtenerse esencialmente de manera simultánea y con intervalo de tiempo mucho menor entre lecturas. La velocidad de conmutación es síncrona con el mensaje de navegación de 50 bps o 20 milisegundos por bit. Una secuencia completa con cuatro satélites se completa en 20 milisegundos o después de 40 milisegundos para los receptores de frecuencia dual.

Aunque los canales de un seguimiento continuo en paralelo son baratos y dan receptores con un buen desempeño, los receptores GPS basados en la tecnología de multiplexación pronto estarán disponibles a un precio más accesible, debido al continuo desarrollo electrónico.

Microprocesador. Para controlar el funcionamiento del receptor GPS, un microprocesador es esencial para recibir y procesar las señales así como para decodificar los mensajes de tipo *broadcast*. Como estos dispositivos se han vuelto más y más poderosos, han absorbido la mayoría del procesamiento de la señal del receptor.

El software o programa del microprocesador del receptor se encuentra en una memoria ya sea internamente en el microprocesador o en circuitos integrados auxiliares que son parte del receptor.

Oscilador de precisión. La frecuencia de referencia es proporcionada por este oscilador. Normalmente osciladores de cuarzo son usados en los receptores desde que la información de precisión de reloj es obtenida de los satélites GPS y el error proporcionado por el oscilador de cuarzo puede ser corregido mediante la técnica de doble diferenciación.

Para navegación con dos o tres satélites si se requiere un oscilador de alta precisión externo.

Fuente de alimentación. Las primeras generaciones de receptores GPS solían consumir mucha energía. Los receptores modernos están diseñados para consumir la menor cantidad de energía posible. Algunos receptores poseen una batería externa recargable de Nickel-Cadmio.

Capacidad de Memoria. Para fines de procesamiento toda la información debe ser almacenada en dispositivos de memoria internos o externos. Los receptores modernos tienen una capacidad de memoria interna de 5 [Mbyte] o más.

Clasificación de los receptores GPS

Los receptores GPS pueden ser divididos en varios grupos dependiendo del criterio que se use para evaluarlos.

Basados en la adquisición de información según la modulación:

- Código C/A
- Código C/A + portadora L1
- Código C/A + portadora L1 + portadora L2
- Código C/A + portadora L1, L2 + código P
- Portadora L1 (no muy común)
- Portadora L1,L2 (raramente usado)

Basados en su uso de los canales GPS:

- Receptores Multicanal
- Receptores Secuencial
- Receptores Multiplexor

Basados en el tipo de usuario:

- Receptores Militares
- Receptores Civiles
- Receptores de Navegación
- Receptores Geodésicos

Especificaciones técnicas de los receptores GPS utilizados

A lo largo de las etapas del proyecto se utilizaron 3 módulos receptores GPS.

1. EM-408
2. Itead Studio GPS shield para Arduino con antenna externa
3. EM-506

Las características de los componentes anteriores serán mencionadas a continuación.

RECEPTOR EM-408

El receptor GPS EM-408 de USGlobalSat está basado en el chipset SiRF StarIII. Incluye una antena de parche basada en microcinta incorporada al módulo y un conector MMCX para una antena externa opcional. Incluye también un cable de conexión de 5 pines.

Especificaciones Técnicas:

Características Generales	
Chipset	SiRF Star III
Frecuencia	L1, 1575.42 [MHz]
Código C/A	1.023[MHz]
Protocolo	Nivel Eléctrico: TTL Voltaje de salida: 0-2.85 [V] Baud rate: 4800-57600 [bps] (ajustable) Mensaje de salida: NMEA 0183 GGA, GSA, GSV, RMC, VTG y GLL
Canales	20
Sensibilidad	-159 [dBm]
Arranque en frío	42 [s] en promedio
Arranque tibio	38 [s] en promedio
Arranque caliente	8 [s] en promedio
Readquisición	0.1 [s] en promedio
Precisión	Posición: 10 [m] 5 [m] con WAAS
Máxima altitud	18,000 [m]
Máxima aceleración	4G
Máxima velocidad	515 [m/s]

Potencia	
Alimentación	3.3 [V]
Consumo	44[mA]

Condiciones ambientales	
Rango de humedad	5% - 9 % sin condensación
Temperatura de operación	-40°C – 85°C

Dimensiones Físicas: 35.4 [mm] x 36.4 [mm] x 8.3 [mm]

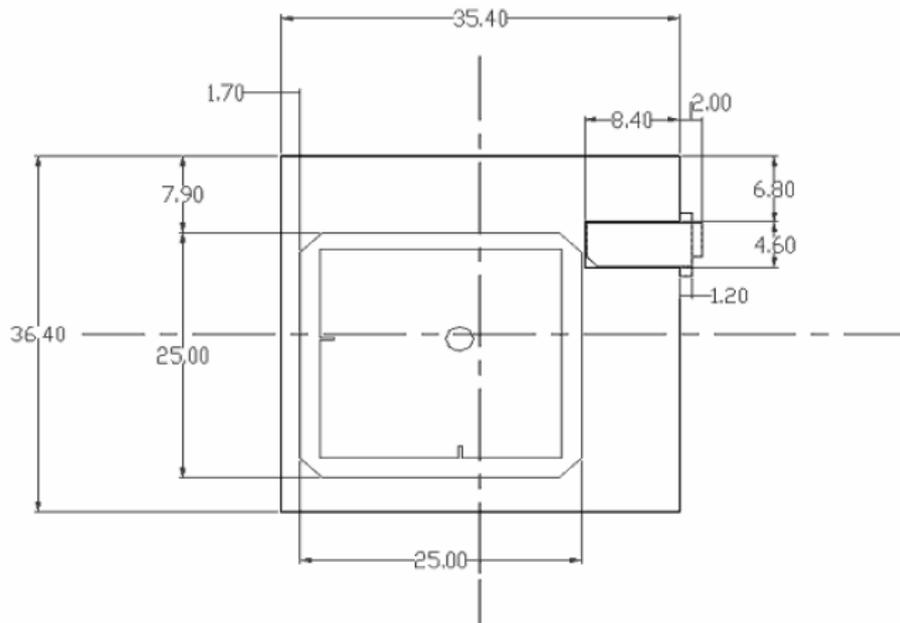


FIGURA 1.8 DIMENSIONES FÍSICAS DEL RECEPTOR EM-408

Conectores verticales de EM-408

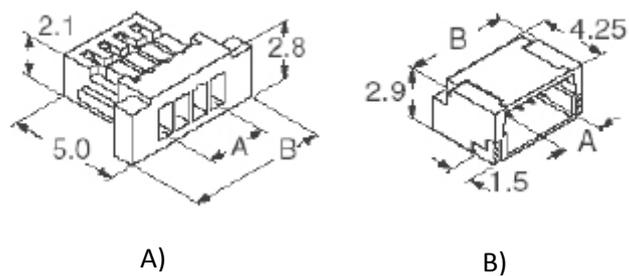


FIGURA 1.9 A) CONECTOR HEMBRA B) CONECTOR MACHO (PCB)

Asignación de Pines del receptor EM-408



FIGURA 1.10 A) RECEPTOR EM-408 B) MAPA DE PINES DEL RECEPTOR EM-408

Pin	Función
1: Activado/Desactivado	Indicador On/Off
2: GND	Tierra del circuito
3: Rx	Canal de recepción de datos GPS
4: Tx	Canal de transmisión de datos GPS
5: VCC	Entrada de alimentación 3.3 [V]

ITEAD STUDIO GPS SHIELD PARA ARDUINO CON ANTENA EXTERNA



FIGURA 1.11 ITEAD STUDIO GPS SHIELD, ARDUINO UNO Y ANTENA EXTERNA

El Itead Studio GPS shield es una tarjeta desarrollada para ser compatible con arduino y otorgarle, funcionalidad de sistema de posicionamiento global además de agregar interfaz micro SD permitiendo almacenar datos de una manera sencilla utilizando una tarjeta micro SD.

Esta tarjeta es compatible con 5 y 3.3 [V] haciéndola compatible con varios modelos de receptores GPS. Además se incluye un conector SMA para conectar una antena externa como fue en el caso de este proyecto.

Las características principales de esta tarjeta son:

- Interfaz micro SD
- Interfaz UART
- Temperatura de operación: -40°C - +85°C
- Baud rate 9600 bps
- Microcontrolador GPS SiRF star III:
 - Receptor de 20 canales
 - Consumo de 62 [mW] durante operación continua
 - Sensibilidad de -159 dBm
 - Funcionalidad (WAAS, MSAS EGNOS)
- Antena externa:
 - Frecuencia de operación 1.57542 [GHz]
 - Tipo externa

Generales de la PCB	
Tamaño de la PCB	55.88[mm] x 54.1[mm] x 1.6[mm]
Indicadores	On/off
Alimentación	Compatible con arduino
Protocolo de comunicación	UART

Potencia			
Especificación	Min	Max	Unidades
Voltaje de alimentación	4.5	5.5	V

Uno TTL	3	5.5	V
Cero TTL	-0.3	0.5	V
Baud rate	38400	38400	Bps

Footprint de la tarjeta:

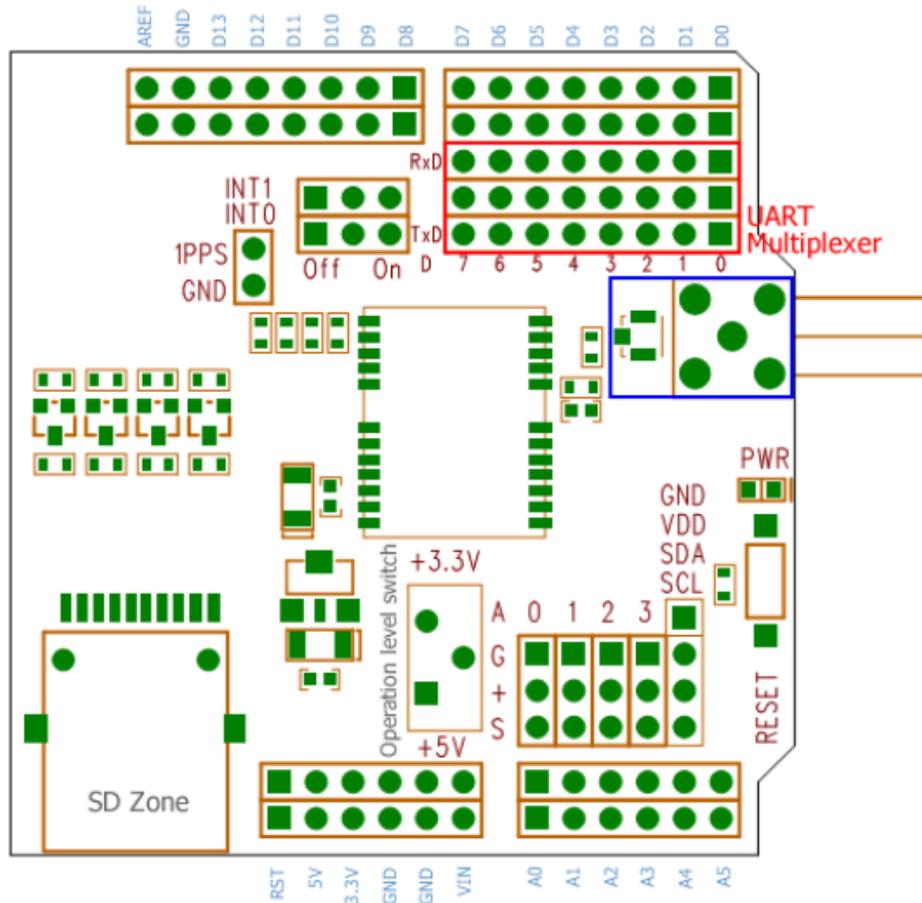


FIGURA 1.12 FOOTPRINT DE LA TARJETA ITEAD STUDIO

Designación de pines con relación a Arduino:

Pin de Arduino	Función
D0	Datos
D1	Din
D10	CSN

D11	MOSI
D12	MISO
D13	SCK
A4	IIC_SDA
A4	IIC_SCL

RECEPTOR EM-506

El receptor GPS EM-506 de USGlobalSat está basado en el chipset SiRF StarIV. Incluye una antena de parche basada en microcinta incorporada al módulo, regulación de voltaje a bordo, LED indicador de estado, RAM respaldada con batería e interfaz de cable de 6 pines.

Este módulo GPS provee una sensibilidad mayor aun estando en entornos con alta densidad de follaje o en zonas urbanas. Hace uso de la tecnología SiRF CGEE (*Client Generated Extended Ephemeris*) con la que predice las posiciones de los satélites hasta de 3 días y eso resulta en un encendido en menos de 15 [s] sin ayuda de ninguna red.



FIGURA 1.13 RECEPTOR EM-506

Especificaciones Técnicas:

Características Generales	
Chipset	SiRF Star IV
Frecuencia	L1, 1575.42 [MHz]
Código C/A	1.023[MHz]
Protocolo	Nivel Eléctrico: TTL Voltaje de salida: 0.4 - 3.3 [V] Baud rate: 4800-57600 [bps] (ajustable) Mensaje de salida: NMEA 0183 GGA, GSA, GSV, RMC, VTG y GLL
Canales	20
Sensibilidad	-163 [dBm]
Arranque en frío	<35 [s], <15[s] con CGEE
Arranque tibio	<35 [s], <15[s] con CGEE
Arranque caliente	1 [s]
Readquisición	0.1 [s] en promedio
Precisión	Posición: 10 [m] 5 [m] con WAAS
Máxima altitud	18,000 [m]
Máxima aceleración	4G
Máxima velocidad	515 [m/s]

Potencia	
Alimentación	5 [V]
Consumo	50[mA], 34[mA] con "fix"

Condiciones ambientales	
Rango de humedad	5% - 9 % sin condensación
Temperatura de operación	-40°C – 85°C

Dimensiones Físicas: 30 [mm] x 30 [mm] x 10.7 [mm]

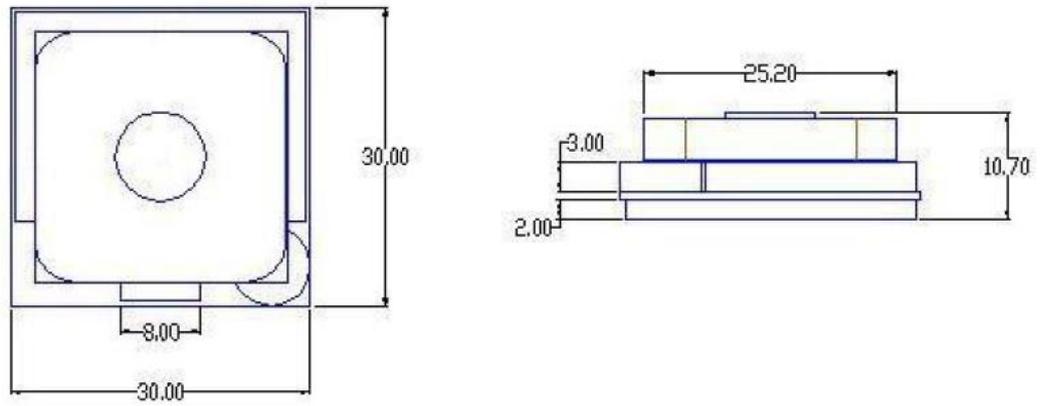


FIGURA 1.14 DIMENSIONES FÍSICAS DEL RECEPTOR EM-506

Conectores verticales de EM-506:

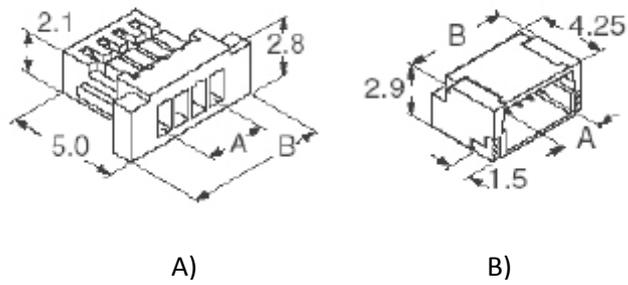


FIGURA 1.15 A) CONECTOR MACHO B) CONECTOR HEMBRA (PCB)

Asignación de Pines del receptor EM-506:

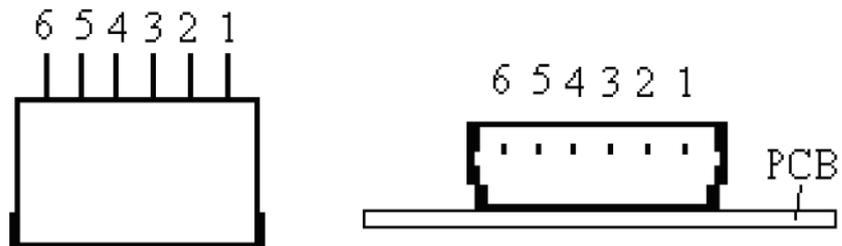


FIGURA 1.16 ASIGNACIÓN DE PINES DEL RECEPTOR EM-506

Pin	Función
1: GND	Tierra del circuito
2: VIN	Alimentación 4.5 - 6.5 [V]
3: Rx	Canal de recepción de datos GPS
4: Tx	Cana de transmisión de datos GPS Salida a 3.3 [V]
5: GND	Tierra del circuito
6: Directive	Indica el estado GPS

Protocolo de comunicaciones NMEA 0183

El estándar NMEA 0183

La National Marine Electronics Association (NMEA) es una asociación no lucrativa de fabricantes, distribuidores, comerciantes, instituciones educativas y otros interesados en ocupaciones con equipo electrónico marítimo periférico. Esta asociación establece el NMEA 0183 que es un estándar industrial voluntario y su primera publicación fue en Marzo de 1983.

Este estándar ha sido actualizado con el paso del tiempo y su última versión publicada en Noviembre del 2008, versión 4.00, está disponible en la página de la NMEA.

Los instrumentos en la norma NMEA 0183 son designados como *talkers* (parlantes) o como *listeners* (oyentes) y en algunos casos algunos instrumentos poseen ambas designaciones, utilizando una interfaz serial asíncrona con los parámetros mostrados en la siguiente tabla:

Baudaje	4800
Número de bits de datos	8 (bit 7 es cero)
Bits de parada	1(o más)
Paridad	Ninguna
Saludo	Ninguno

Interfaz eléctrica

La norma NMEA 0183 permite un *talker* y varios *listeners* sobre un circuito. El cableado de interconexión recomendado es un par trenzado blindado, con el blindaje solo aterrizado en el talker. El estándar no especifica el uso de un conector en particular.

Es muy recomendable que la salida del *talker* cumpla con el EIA RS-422, un sistema diferencial con dos líneas de señal, "A" y "B". Las señales de control diferencial no tienen referencia a tierra y tienen una mejor inmunidad al ruido. Una sola línea con nivel TTL también se acepta. Los voltajes de la línea A corresponden a aquellos en el único cable TTL, mientras que los voltajes en B son invertidos (si la salida A es +5[V] la salida B es 0[V]). Esta es la operación unipolar RS-422. En modo bipolar se utilizan +/- 5[V].

En ambos casos, el circuito receptor recomendado utiliza un optoacoplador con circuitos de protección. La entrada debe aislarse de la tierra del receptor. En la práctica, el cable sencillo, o el cable A del RS-422 puede ser conectado directamente a la entrada RS-232 de una computadora. De hecho, algunos de los receptores como por ejemplo los receptores GPS de bolsillo no tienen una salida diferencial RS-422, solo tienen una línea sencilla con un nivel de señal TTL o CMOS compatible.

Formato general de los mensajes

En la norma NMEA 0183 todos los datos son transmitidos como mensajes. Únicamente se permiten caracteres ASCII desplegados en la pantalla, más <CR> (*carriage return*) y <LF> (*line feed*). Cada mensaje comienza con el signo \$ y termina con <CR><LF>.

Existen tres tipos básicos de mensajes:

- Mensajes de parlante (*talker sentences*)
- Mensajes de propietario (*proprietary sentences*)
- Mensajes de petición (*query sentences*)

Para el sistema presentado, los mensajes más relevantes son los mensajes de parlante, estos son, los mensajes de navegación que manda la constelación de satélites GPS.

El formato general de los mensajes parlante es:

\$tsssd1,d2,...<CR><LF>

Las primeras dos letras que siguen a "\$" forman el identificador de parlante (*talker identifier*). Los siguientes tres caracteres, representados por "sss", forman el identificador de mensaje (*sentence identifier*), y son seguidos por un número variable de campos de datos separados por comas, seguidos por un *checksum* opcional, y finalmente el mensaje termina con <CR><LF>.

Los campos de datos son definidos de manera única para cada tipo de mensaje. Un mensaje puede consistir de más de 80 caracteres, aparte de "\$", <CR> y <LF>. Si los datos de alguno de los campos no están disponibles, ese campo será omitido, pero las comas que lo delimitan serán transmitidas sin ningún espacio entre ellas.

Formato del mensaje GPS

El sistema de posicionamiento global por satélite GPS utiliza el identificador de parlante "GP" y el contenido de los mensajes dependerá del tipo de mensaje del satélite.

Para el sistema GPS los identificadores de mensaje NMEA 0183 son:

- **GGA** *Global Positioning System Fix Data*
- **GLL** *Latitude/Longitude*
- **GSA** *GNSS DOP and Active Satellites*
- **GSV** *GNSS Satellites in View*
- **RMC** *Recommended Minimum Specific GNSS Data*
- **VTG** *Course Over Ground and Ground Speed*

Los receptores con los que se trabajó soportan los 6 tipos de mensajes antes mencionados pero en este proyecto utilizaremos únicamente el mensaje GGA para extraer la información deseada.

El formato del mensaje GGA es el siguiente:

\$GPGGA,hhmmss.ss,ddmm.mm,a,dddmm.mm,a,x,xx,x.x,x.x,M,x.x,M,x.x,xxxx*hh

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Los 15 campos van seguidos de <CR><LF> como anteriormente se mencionó. Los campos se definen de la manera siguiente:

1. Tiempo UTC
2. Latitud
3. N o S (Norte o Sur)
4. Longitud
5. E o W (Este u Oeste)
6. Indicador de estado GPS (0=datos no disponibles, 1=GPS fijo, 2=DGPS fijo)
7. Número de satélites en vista (00 a 12)
8. Dilución horizontal de precisión
9. Altitud de antena sobre/bajo el nivel del mar
10. Altitud en unidades de antena, metros
11. Separación Geoidal
12. Unidades de separación Geoidal
13. Datos de edad de GPS diferencial
14. Identificación de estación de referencia diferencial
15. Checksum

De la cadena de datos GGA se obtendrán los datos que sean requeridos para este proyecto como el tiempo UTC, latitud y longitud principalmente.

Capítulo 2 El Microprocesador ATmega328

El microprocesador

Los microprocesadores se empezaron a utilizar en algunos productos desplazando la lógica discreta, remplazando a los sistemas que utilizaban flip-flops, compuertas lógicas, contadores, y otros componentes TTL.

Desde el punto de vista lógico, singular y funcional, el microprocesador está compuesto básicamente por: varios registros, una unidad de control, una unidad aritmético lógica, y dependiendo del procesador, una unidad de punto flotante.

El microprocesador ejecuta instrucciones almacenadas como números binarios organizados secuencialmente en la memoria principal. La ejecución de las instrucciones se puede realizar en varias fases:

- *Prefetch*, prelectura de la instrucción desde la memoria principal
- *Fetch*, envío de la instrucción al decodificador
- Decodificación de la instrucción, es decir, determinar qué instrucción es y por tanto qué se debe hacer
- Lectura de operandos (si los hay)
- Ejecución, lanzamiento de las máquinas de estado que llevan a cabo el procesamiento
- Escritura de los resultados en la memoria principal o en los registros

Algunas de las ventajas de los microprocesadores sobre los diseños discretos son:

- Disminución de circuitos integrados, tarjetas impresas y conectores por lo que el costo de producción y ensamblaje se reduce
- Menores requerimientos de potencia
- Disminución de interconexiones lo que provoca mayor confiabilidad
- Simplificación en evaluación, prueba y rediseño del sistema. Su operación es fácilmente modificable ya que únicamente se necesita cambiar el programa o remplazar la memoria en lugar de cambiar todo el cableado de un sistema de lógica discreta.

Todas las microcomputadoras y los microcontroladores, a pesar de variar en su arquitectura, tienen en común el microprocesador.

Normalmente el microprocesador se encuentra en un solo chip de alta escala de integración que contiene la unidad de control y la Unidad Lógico Aritmética.

Algunas de las funciones que realiza el microprocesador son:

- Proveer las señales de control y señales de tiempo
- Cargar los datos y comandos de la memoria
- Transferencia de datos a los instrumentos de entrada y salida
- Decodificar las instrucciones
- Realizar las operaciones aritméticas y lógicas requeridas por las instrucciones
- Responder a las señales de control de entrada o salida como RESET

El microprocesador contiene toda la circuitería lógica para desarrollar esas funciones, pero esta lógica interna del microprocesador no es accesible externamente. La circuitería interna del microprocesador solo puede modificarse mediante el programa que se guarda en su memoria, para que el microprocesador lo ejecute.

Gracias a lo anterior el microprocesador es tan flexible y versátil.

Características técnicas del microprocesador ATmega328

El ATmega328 es un microcontrolador de 8 bits basado en tecnología CMOS de bajo consumo de energía. Este microcontrolador está basado en la arquitectura RISC. Ejecutando instrucciones en tan solo un ciclo de reloj el ATmega328 permite lograr un rendimiento cerca de 1 MIPS por MHz permitiendo al sistema diseñado, optimizar el consumo de potencia contra velocidad de procesamiento.

El microcontrolador ATmega328 fue utilizado para este proyecto ya que nos permitió reutilizar el archivo .ino con el que se controló al receptor GPS.

Otro factor importante para la decisión de selección fue el consumo de potencia ya que se busca realizar un diseño de bajo consumo de potencia y con alto rendimiento y el ATmega328 cumple todos los requerimientos de este proyecto.

Diagrama a bloques del microcontrolador ATmega328

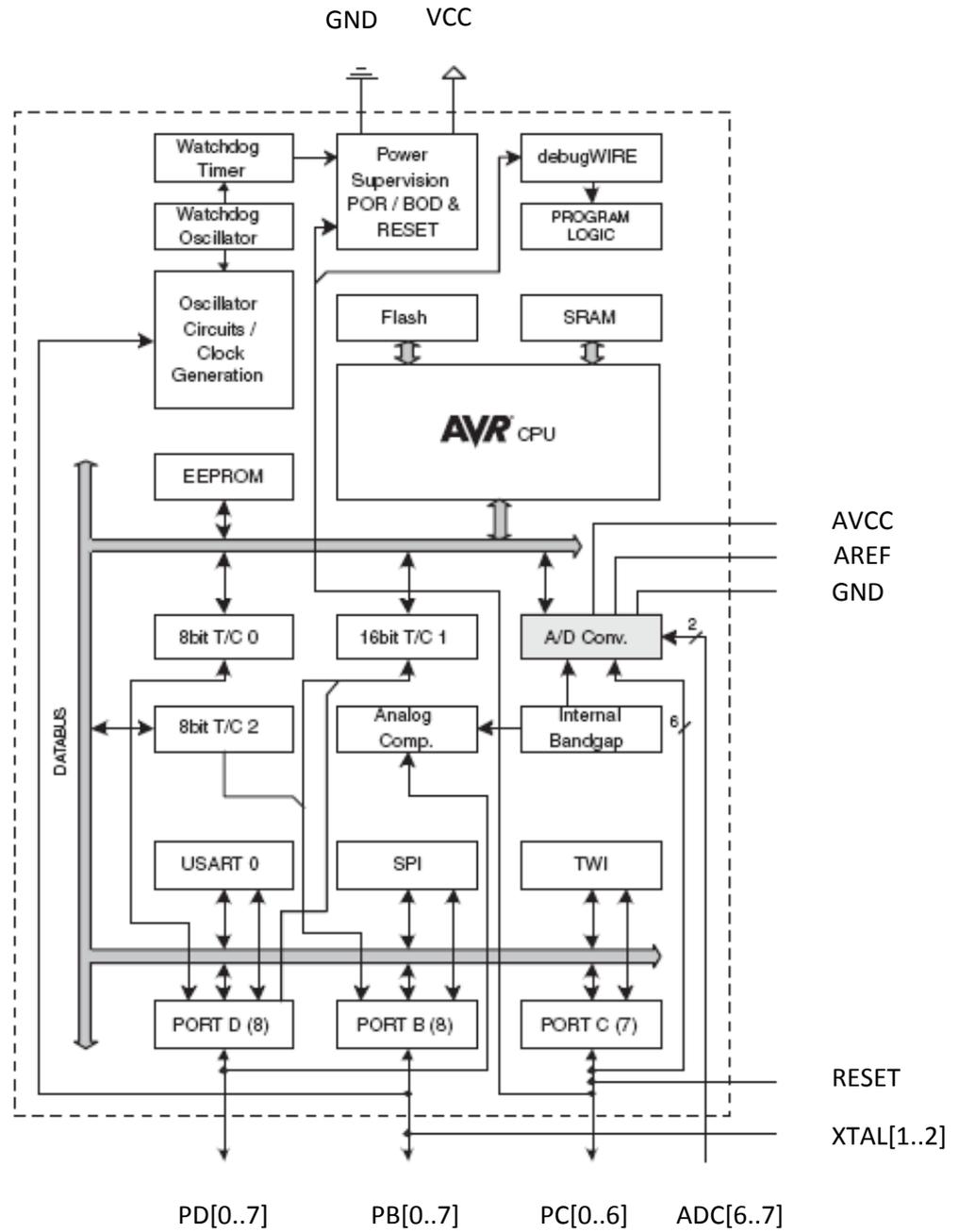


FIGURA 2.1 DIAGRAMA A BLOQUES DEL MICROCONTROLADOR ATMEGA328

Asignación de pines para configuración de 28 pines

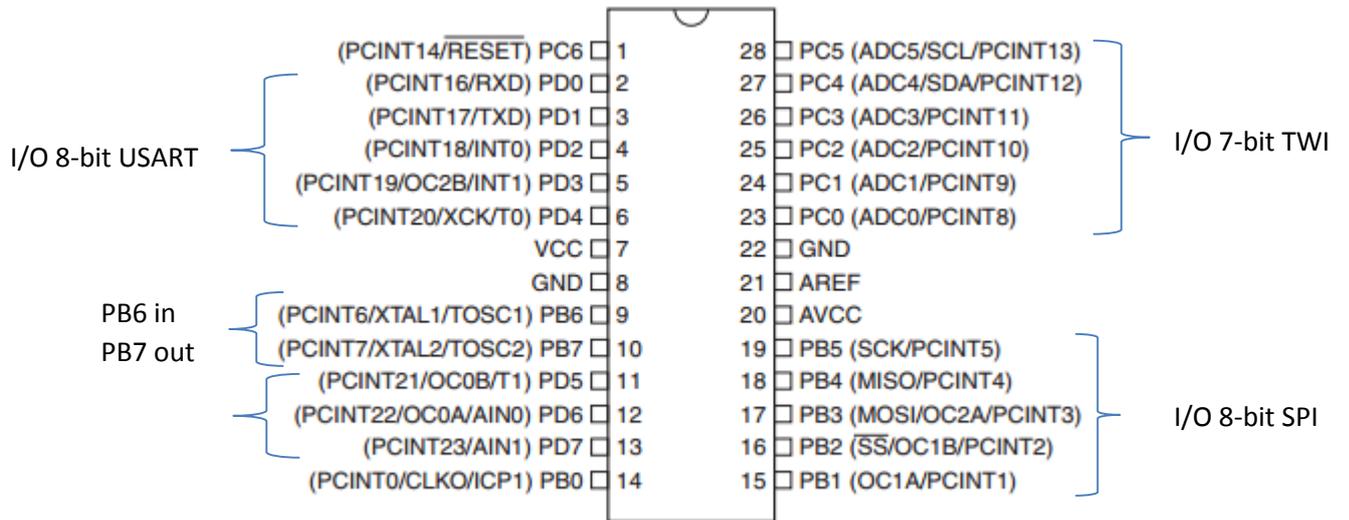


FIGURA 2.2 MAPA DE PINES DEL MICROCONTROLADOR ATMEGA328 VERSIÓN DE EMPAQUETADO 28 PINES

Especificaciones de algunos pines

- **Vcc**

Entrada digital 1.8 - 5.5 [V]

- **PC6 Reset input**

Si un pulso mayor al tiempo requerido en valor bajo es detectado, se producirá un reseteo.

- **AVcc**

Fuente de alimentación del convertidor A/D (PC 3:0). Debe ser externamente conectado a Vcc incluso si el ADC no es usado.

Si el ADC se usa, debe ser conectado a Vcc con un filtro paso bajas.

- **AREF**

Pin de referencia analógica para el convertidor A/D.

Funciones de los puertos del microcontrolador ATmega328

Funciones de los puertos B

Pin	Funciones
PB7	XTAL2 (Chip Clock Oscillator pin 2) TOSC2 (timer Oscillator pin 2) PCINT7 (Pin Change Interrupt 7)
PB6	XTAL1 (Chip Clock Oscillator pin 1) TOSC1 (timer Oscillator pin 1) PCINT6 (Pin Change Interrupt 6)
PB5	SCK (SPI Bus Master clock Input) PCINT5 (Pin Change Interrupt 5)
PB4	MISO (Bus SPI Master Input/Slave Output) PCINT4 (Pin Change Interrupt 4)
PB3	MOSI (bus SPI Master Output/Slave Input) OC2A (Timer/Counter2 Output Compare Match A Output) PCINT3 (Pin Change Interrupt 3)
PB2	SS (Bus SPI Master Slave select) OC1B (Timer/Counter1 Output Compare Match B Output) PCINT2 (Pin Change Interrupt 2)
PB1	OC1A (Timer/Counter1 Output Compare Match A Output) PCINT1 (Pin Change Interrupt 1)
PB0	ICP1 (Timer/Counter1 Input Capture Input) CLK0 (Divided System Clock Output) PCINT0 (Pin Change Interrupt 0)

Funciones de los puertos C

Pin	Funciones
PC6	RESET (pin de reinicio) PCINT14 (Pin Change Interrupt 14)
PC5	ADC5 (ADC Input Channel 5) SCL (2-wire Serial Bus Clock Line) PCINT13 (Pin Change Interrupt 13)
PC4	ADC4 (ADC Input Channel 4) SDA (2-wire Serial Bus Data Input/Output Line)

	PCINT12 (Pin Change Interrupt 12)
PC3	ADC3 (ADC Input Channel 3) PCINT11 (Pin Change Interrupt 11)
PC2	ADC2 (ADC Input Channel 2) PCINT10 (Pin Change Interrupt 10)
PC1	ADC1 (ADC Input Channel 1) PCINT9 (Pin Change Interrupt 9)
PC0	ADC0 (ADC Input Channel 0) PCINT8 (Pin Change Interrupt 8)

Funciones de los puertos D

Pin	Funciones
PD7	AIN1 (Analog Comparator Negative Input) PCINT23 (Pin Change Interrupt 23)
PD6	AIN0 (Analog Comparator Positive Input) OCPA (Timer/Counter0 Output Compare Match A Output) PCINT22 (Pin Change Interrupt 22)
PD5	T1 (Timer/Counter 1 External Counter Input) OCOB (Timer/Counter0 Output Compare Match B Output) PCINT21 (Pin Change Interrupt 21)
PD4	XCK (USART External Clock Input/Output) T0 (Timer/Counter 0 External Counter Input) PCINT20 (Pin Change Interrupt 20)
PD3	INT1 (External Interrupt 1 Input) OC2B (Timer/Counter2 Output Compare Match B Output) PCINT19 (Pin Change Interrupt 19)
PD2	INT0 (External Interrupt 0 Input) PCINT18 (Pin Change Interrupt 18)
PD1	TXD (USART Output pin) PCINT17 (Pin Change Interrupt 17)
PD0	RXD (USART Input pin) PCINT16 (Pin Change Interrupt 16)

Relación de pines ATmega328 con Arduino

En la siguiente imagen se puede identificar cada uno de los pines del microcontrolador ATmega328 relacionándolos con las funciones de la tarjeta Arduino UNO.

Esta relación resultó crucial para pasar de un sistema GPS que funciona con la tarjeta de prueba ARDUINO a un sistema GPS con los componentes y conexiones mínimas requeridas.

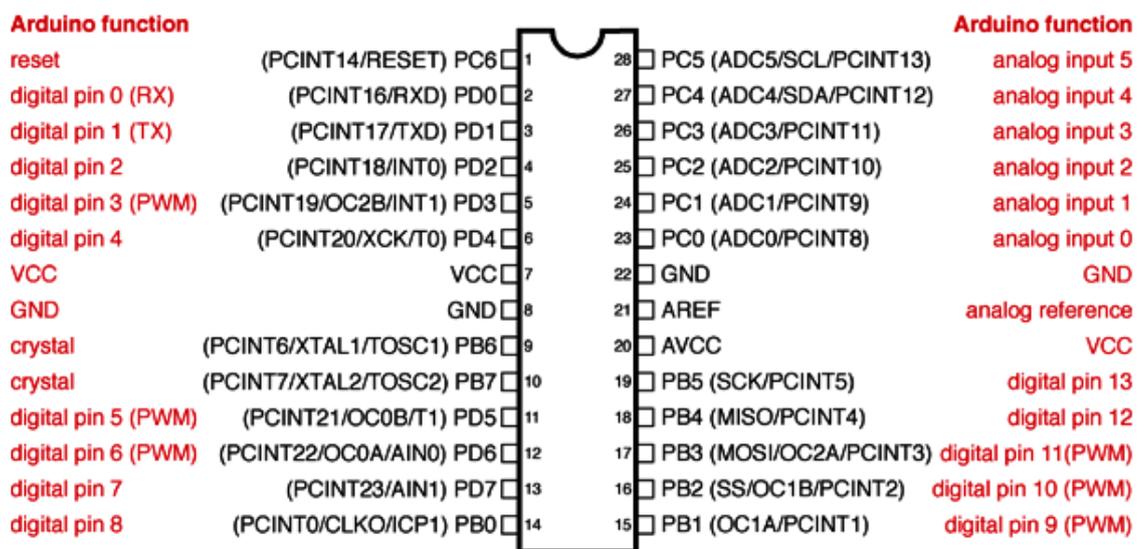


FIGURA 2.3 RELACIÓN DE PINES DE MICROCONTROLADOR ATMEGA328 CON PINES ARDUINO UNO

Pines Digitales 11, 12 y 13 son usados por el *header* ICSP para realizar las conexiones MISO, MOSI y SCK. Evitar cargas de baja impedancia cuando el *header* ICSP está en uso.

Capítulo 3 Sistemas receptores GPS

Sistema con módulo receptor EM-408

Este sistema consta de los siguientes componentes:

- Arduino UNO
- Shield GPS de Sparkfun
- Modulo receptor EM-408

Para realizar la conexión entre el modulo receptor y la tarjeta de desarrollo arduino es necesario contar con un medio adaptador ya que el cable conector del módulo receptor no cuenta con las medidas necesarias y compatibles con los headers de la tarjeta de desarrollo arduino.

Para resolver la problemática anterior se propuso utilizar el shield GPS de Sparkfun.

Las piezas para armar el shield GPS se muestran en la figura 3.1

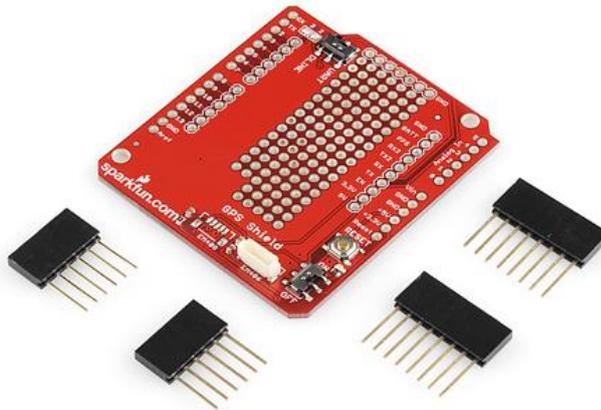


FIGURA 3.1 PIEZAS INCLUIDAS EN EL KIT DEL SHIELD GPS DE SPARKFUN

- 1 x GPS Shield PCB
- 2 x 8-pin de Arduino apilable Header
- 2 x 6-pin de Arduino apilable Header

Para tener funcionando correctamente el equipo, se necesitó soldar los headers en el Shield GPS como se aprecia en la figura 3.2

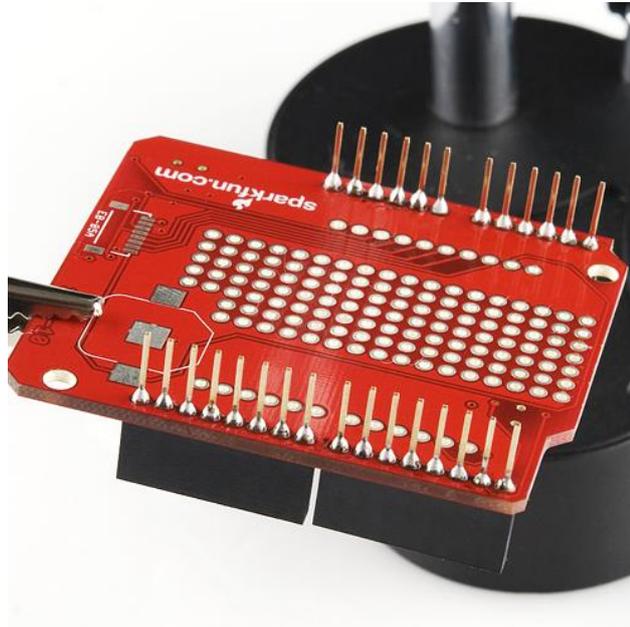


FIGURA 3.2 SHIELD GPS SPARKFUN CON SUS RESPECTIVOS HEADERS SOLDADOS DE MANERA CORRECTA

Además de lo anterior se soldó el conector vertical SMD compatible con el módulo EM-408 mostrado en la figura 3.3 ya que el shield únicamente incluye un conector vertical compatible con el módulo receptor EM-406(ver figura 3.4).



FIGURA 3.3 CONECTOR VERTICAL SMD COMPATIBLE CON RECEPTOR EM-408

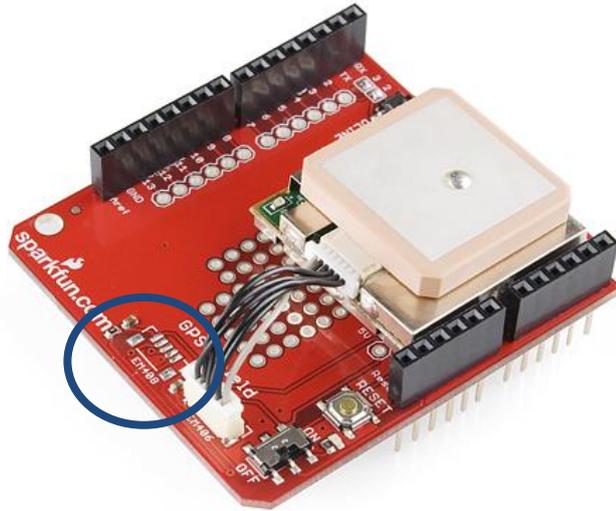


FIGURA 3.4 SHIELD GPS DE SPARKFUN CONECTADO A UN MÓDULO RECEPTOR EM-406

Ya con el shield GPS ensamblado correctamente se ensambla con la tarjeta de arduino y el modulo GPS a utilizar ya sea en su versión EM-406 o Em-408.

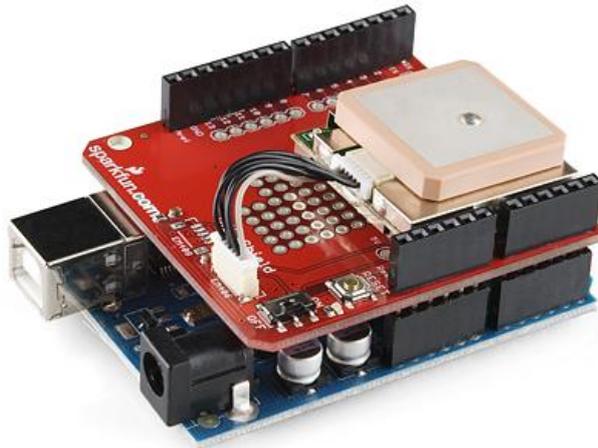


FIGURA 3.5 MODULO RECEPTOR GPS ENSAMBLADO CON SHIELD GPS SPARKFUN Y ARDUINO UNO

Para realizar una selección del módulo receptor a utilizar, se analizaron las características principales de los equipos lo cual puede visualizarse en la tabla 3.1

Módulo	EM-406	EM-408
Tipo de Antena	Chip integrado	Chip integrado
Dimensiones	30x30x10.5 [mm]	36.4x35.4x8.3 [mm]
Tiempo de iniciación	Frio: 42 [s] Caliente: 1 [s]	Frio: 42 [s] Caliente: 8 [s]
# de Canales	20	20
Protocolo	NMEA 0183 y SiRF binario	NMEA 0183 y SiRF binario
Tasa de actualización	1 [Hz]	1 [Hz]
Requerimientos de alimentación	4.5-6.5 [V] 70 [mA]	3.3 [V] 75 [mA]
Sensibilidad	-159 [dBm]	-159 [dBm]
Ventajas	Alta precisión Compatibilidad con arduino Muy compacto Incluye antena LED indicador Más económico	Alta precisión Compatibilidad con arduino Compacto Incluye antena Conexión para antena externa 3.3 [V] para energizar
Desventajas	Antena integrada 5[V] para energizar Uso de conectores SMD	Uso de conectores SMD Ligero aumento de dimensiones a causa de conector para antena externa

TABLA 3.1 COMPARACIÓN DE MÓDULOS RECEPTORES EM-406 Y EM-408

En base a lo anterior se decidió trabajar con el módulo EM-408 tomando dos criterios principales:

- 1) Energización. No se conocen las condiciones en las que se encontrará el sistema por lo que es importante que consuma la menor cantidad de energía posible.
- 2) Tipo de antena. Los dos receptores cuentan con una antena integrada, sin embargo el receptor EM-408 cuenta con un conector para antena externa lo que podría representar una ventaja frente al otro módulo receptor.

Contando con el módulo receptor GPS EM-408, el shield GPS de sparkfun y la tarjeta de desarrollo arduino se prosiguió a realizar el programa extensión .ino con el que el microcontrolador realiza las tareas de control, adquisición e

interpretación de los datos obtenidos con el receptor GPS. Este programa puede encontrarse comentado en la sección de Anexos.

Las pruebas realizadas con este sistema están ubicadas en el capítulo 5 Pruebas y Resultados.

Sistema con módulo receptor EM-408, arduino MEGA y compatibilidad micro SD

Este sistema consta de los siguientes componentes:

- Arduino MEGA
- Wireless SD Shield
- Módulo receptor EM-408
- Tarjeta micro SD

Esta configuración del sistema receptor GPS fue considerada para otorgar la funcionalidad de almacenamiento de datos de ser deseado en etapas posteriores del proyecto.

Es por eso que se hace uso del Wireless SD shield de Arduino.

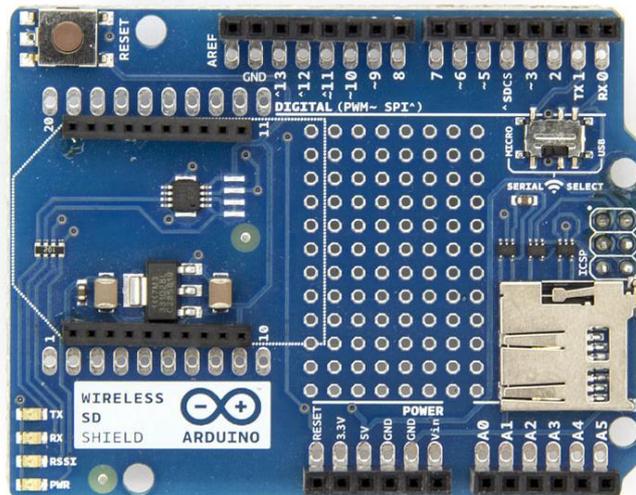


FIGURA 3.6 WIRELESS SD SHIELD DE ARDUINO

Este shield tiene incorporado un sujetador de tarjeta micro SD la que se utilizará para dar la capacidad de almacenamiento de datos a nuestro sistema.

La tarjeta de desarrollo Arduino se comunica de manera serial con el shield Wireless SD por lo que será necesario utilizar los pines digitales 0 para recepción de datos y 1 para transmitir datos.

Por otro lado el módulo receptor EM-408 también se comunica de manera serial con la tarjeta de desarrollo arduino utilizando igualmente los pines digitales 0 para recepción de datos y 1 para transmitir datos.

Lo anterior supone un problema puesto que si se realizaran las conexiones descritas anteriormente, existiría una colisión de datos entre el receptor EM-408 y el shield Wireless SD.

Para resolver la problemática anterior se propuso utilizar la tarjeta de desarrollo arduino MEGA ya que esta, a diferencia del arduino UNO, cuenta con tres puertos de comunicación serial.

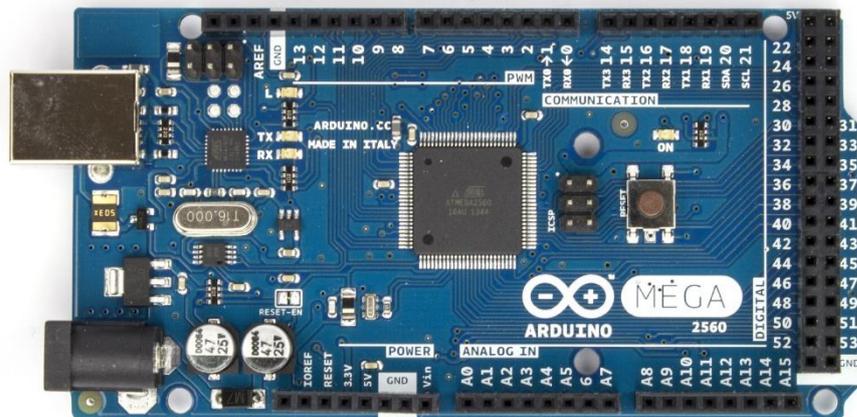


FIGURA 3.7 ARDUINO MEGA

Para el correcto funcionamiento de este sistema se conectó el shield Wireless SD al arduino mega utilizando los pines digitales 0 y 1 para la comunicación serial y el módulo receptor EM-408 fue energizado y conectado a los pines digitales 14 y 15 para transmisión y recepción respectivamente de la información obtenida

por el receptor. Las conexiones requeridas descritas anteriormente se pueden observar en la figura 3.8

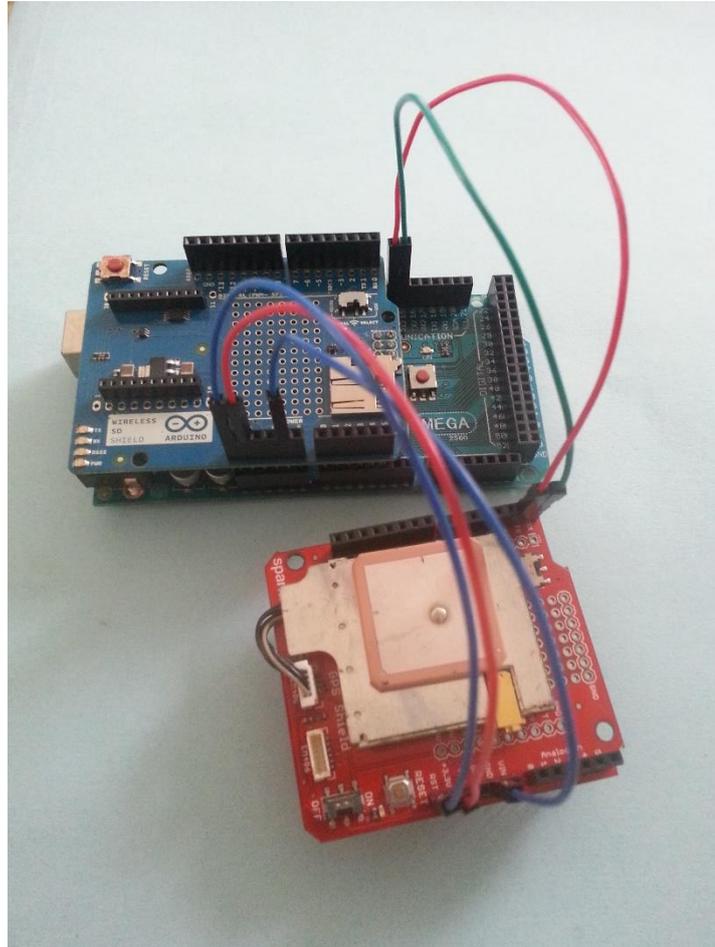


FIGURA 3.8 ARDUINO MEGA CONECTADO A WIRELESS SD SHIELD Y MÓDULO RECEPTOR EM-408

El programa con extensión .ino utilizado para este sistema se encuentra en la sección Anexos.

Las pruebas realizadas con este sistema están ubicadas en el capítulo 5 Pruebas y Resultados.

Sistema con Shield iTead Studio y antena externa

Este sistema consta de los siguientes componentes:

- Arduino UNO
- Shield iTead Studio
- Antena Externa

Debido a la dificultad de conseguir una antena externa compatible con el conector MMX del módulo EM-408 se decidió realizar pruebas con el Shield de iTead Studio ya al igual que el módulo Em-408 incorpora un chipset SiRF III otorgándole características similares e incorpora directamente un conector SMA compatible con la mayoría de las antenas externas disponibles en el mercado.



FIGURA 3.9 ANTENA GPS 3 [V] CON CONECTOR SMA

Otra de las razones por las que se buscó realizar pruebas con este shield es que se deseaba tener un sistema que no estuviera limitado por la ubicación de la antena. Para que el receptor GPS logre determinar la posición del mismo, es

necesario tener en línea de vista al menos 4 satélites y esto puede dejar de cumplirse en función de donde está ubicado el módulo receptor.

Con la antena externa se resuelve el problema anterior ya que sin importar donde se encuentra nuestro módulo receptor, el sistema tendrá la posibilidad de determinar la posición siempre que la antena externa este colocada en un lugar sin demasiadas obstrucciones para lograr tener en línea de vista los 4 satélites requeridos.

Ya que se utilizó la misma tarjeta de desarrollo, arduino UNO, se pudo reutilizar prácticamente todo el programa instalado en el microcontrolador del arduino.

La única modificación requerida es la velocidad a la que trabaja el receptor pues el módulo EM-408 funciona a 4800 baudios y el receptor del shield iTead Studio funciona a 9600 baudios.

El programa utilizado con este receptor extensión .ino puede encontrarse en la sección de Anexos.



FIGURA 3.10 SHIELD ITEAD STUDIO CON ANTENA GPS EXTERNA CONECTADO A ARDUINO UNO

Las pruebas realizadas con este sistema están ubicadas en el capítulo 5 Pruebas y Resultados.

Capítulo 4 Integración del Sistema Simplificado

En el capítulo anterior se muestran tres sistemas cada uno con características diferentes los cuales se retomarán en este capítulo para proponer un sistema simplificado en el que únicamente se tengan los elementos y conexiones mínimas para cumplir los objetivos de este proyecto.

Conocer las características de funcionamiento, programación y especificaciones técnicas de los sistemas anteriores resultó vital para el desarrollo e implementación del sistema simplificado.

Partimos de un sistema el cual trabajó con los siguientes componentes:

- Tarjeta de desarrollo Arduino Uno R3 con ATmega328
- iTeed Shield para arduino con chipset SIRFIII e interfaz SD o GPS shield con receptor EM-408
- Antena externa SMA o antena interna

A lo largo del capítulo se analizará el sistema desde el punto de vista de sus componentes principales:

- Microprocesador
- Receptor GPS
- Interfaz micro SD
- Energización del sistema

Se mostrará el diagrama de bloques del sistema y finalmente con la información anterior se propondrá el diseño de una PCB la que se podrá integrar a un sistema de monitoreo otorgándole la funcionalidad de geo localización.

Finalmente en el capítulo 5 Pruebas y Resultados se implementará un prototipo con el que se validará el diseño de la PCB propuesto.

Microprocesador

Primeramente se analizó la tarjeta de desarrollo Arduino Uno visualizando su esquemático.

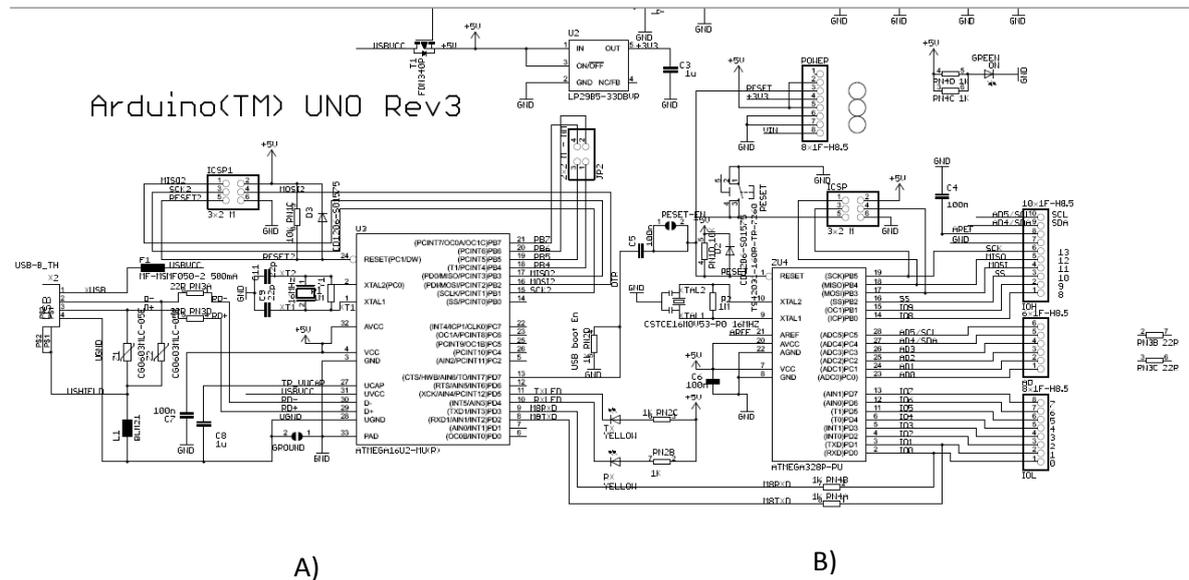


FIGURA 4.1 ESQUEMÁTICO DE TARJETA ARDUINO UNO R3

En la figura 4.1 observamos que la tarjeta de desarrollo posee dos componentes principales, que son los microcontroladores:

- A) ATmega16U2
- B) ATmega328

Y se identificó la principal función de cada uno. Por un lado el microcontrolador donde cargamos nuestro programa en lenguaje .ino es el ATmega328 y por el otro tenemos el microcontrolador ATmega16U2 que es utilizado para realizar la transferencia de información entre la computadora y nuestra tarjeta de desarrollo convirtiendo de interfaz 232 a USB y viceversa.

Teniendo en cuenta lo anterior surge la siguiente pregunta. ¿Cuál es la mínima conexión para el correcto funcionamiento del microprocesador ATmega328?

Una de las principales características del sistema es las dimensiones del mismo. Entre más compacto sea el sistema más fácil se podrá adaptar al sistema de monitoreo. Además, pensando en la economía del proyecto no sería viable colocar una tarjeta arduino UNO por cada sistema receptor que se desee implementar en cada elemento del sistema de monitoreo.

Por lo anterior se buscó entonces, la configuración mínima para el correcto funcionamiento del microcontrolador ATmega328.

Basados en el datasheet del microprocesador ATmega328 comenzamos por definir la función de cada uno de los pines.

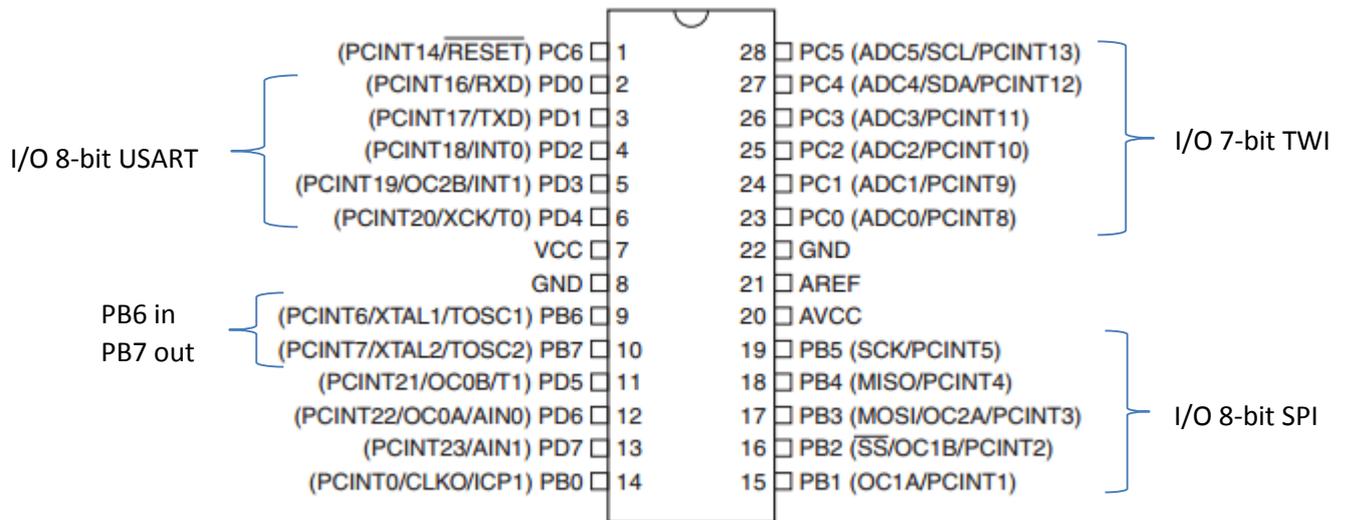


FIGURA 4.2 MAPA DE PINES DEL MICROCONTROLADOR ATMEGA328 VERSIÓN DE EMPAQUETADO 28 PINES

- **Vcc**

Entrada digital 1.8 - 5.5 [V]

- **PC6 Reset input**

Si un pulso mayor al tiempo requerido en valor bajo es detectado, se producirá un reseteo.

- **AVcc**

Fuente de alimentación del convertidor A/D (PC 3:0). Debe ser externamente conectado a Vcc incluso si el ADC no es usado. Si el ADC se usa, debe ser conectado a Vcc con un filtro paso bajas.

Después se realizó una identificación de los pines de la tarjeta Arduino Uno relacionando los pines del microprocesador ATmega328 para simplificar futuras tareas puesto que ya se tiene un sistema funcional con la tarjeta Arduino Uno.

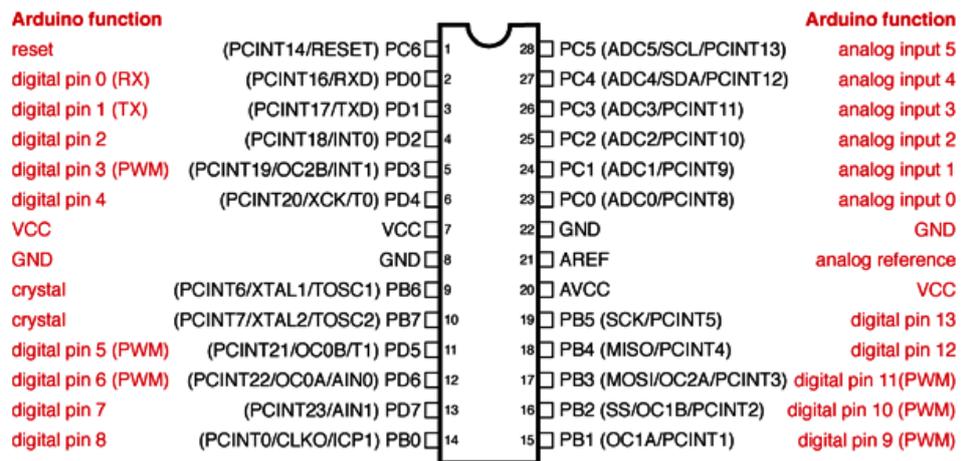


FIGURA 4.23 RELACIÓN DE PINES DE MICROCONTROLADOR ATMEGA328 CON PINES ARDUINO UNO

Acorde con lo anterior se propone el siguiente diagrama de conexión mínima para el microcontrolador ATmega328 en el que además del microcontrolador se necesita el material siguiente:

- 1 cristal de 16 [MHz]
- 2 capacitores cerámicos de 22[pF]
- 1 resistencia de 10[KΩ]

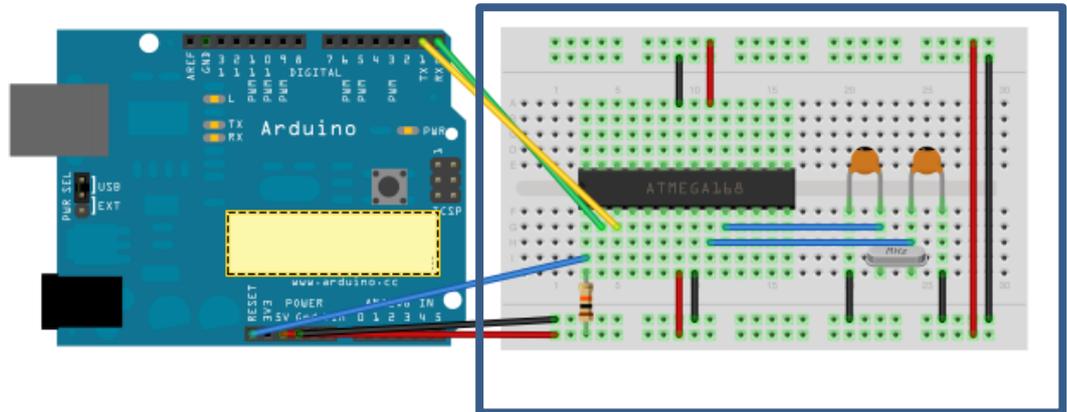


FIGURA 4.4 CONEXIÓN MÍNIMA DE MICROCONTROLADOR ATMEGA328 CONECTADO A ARDUINO UNO

En el diagrama anterior se observan tres conexiones adicionales hacia una tarjeta de desarrollo Arduino Uno sin microcontrolador ATmega328 o ATmega168, que justamente este será la manera en que nosotros cargaremos nuestro programa de tipo .ino al microcontrolador ATmega328 que se encuentra con la mínima conexión.

En la hoja de especificaciones del ATmega328 se menciona que para un correcto funcionamiento del mismo, se deberá inicializar previamente el gestor de arranque (bootloader).

Lo anterior no es necesario cuando se cuenta con la tarjeta de desarrollo Arduino puesto que este gestor ha sido cargado previamente. Para nuestro ATmega328 con la mínima conexión es necesario realizar el grabado del gestor de arranque.

Para realizar el grabado del gestor de arranque del microprocesador se deberá realizar las conexiones que se muestran en el diagrama siguiente además de seguir los pasos que se describirán también a continuación.

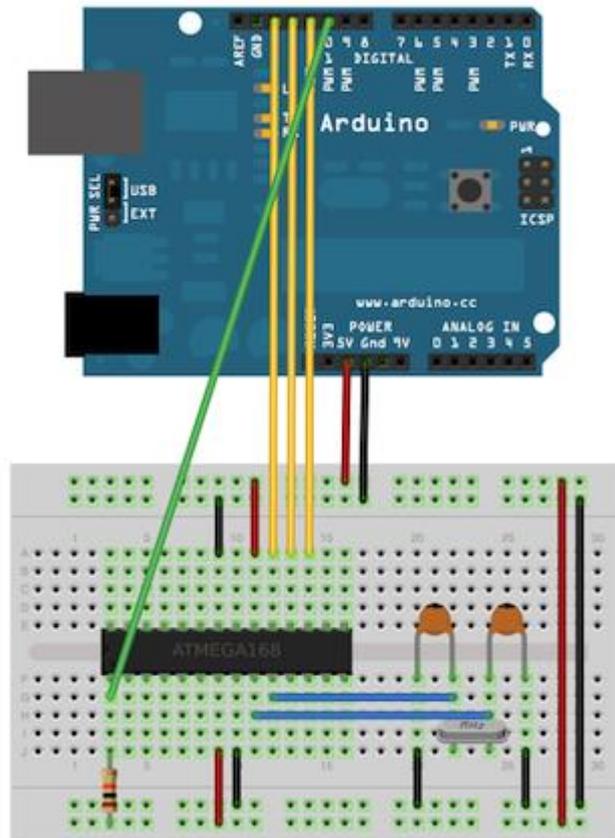


FIGURA 4.5 CONEXIONES NECESARIAS PARA GRABAR EL GESTOR DE ARRANQUE DEL MICROCONTROLADOR ATMEGA328

1. Cargar el programa Arduino ISP a la tarjeta de desarrollo Arduino Uno, seleccionando el puerto y la tarjeta indicada en el IDE de arduino.
2. Realizar las conexiones mostradas
3. Seleccionar Arduino Duemilanove w/ ATmega328 en el menú Herramientas del IDE Arduino.
4. Ejecutar Arduino como ISP(in-system program) desde el Menu Herramientas, Burn Bootloader del IDE de Arduino.

Solo se requiere realizar el procedimiento anterior una vez por cada nuevo microcontrolador ATmega328.

Receptor GPS

Teniendo la configuración mínima del microcontrolador ATmega328 continuamos con el análisis de del dispositivo GPS que utilizaremos en nuestro sistema simplificado.

Los receptores con los que se trabajó anteriormente son:

- Itead Studio GPS shield para Arduino
- EM-408

De acuerdo con la experiencia adquirida al trabajar con los componentes anteriores podemos decir que ambos tienen ventajas y desventajas. Por un lado el EM-408 es más sencillo de utilizar puesto que el chipset SIRFIII está integrado en el receptor mientras que el Itead Studio GPS shield también hace uso de un chipset SIRF III pero de montaje superficial, teniendo un grado de complejidad de implementación, sin embargo, la opción de Itead GPS shield posee compatibilidad con interfaz micro SD para almacenar datos de una forma sencilla y no muy costosa mientras que el receptor EM-408 no cuenta con esa opción.

Otra ventaja del Itead GPS shield sobre el receptor EM-408 es su facilidad de utilizar una antena externa para la recepción de las cadenas de datos GPS.

Bajo la comparación anterior el Itead GPS shield parecía la mejor opción, sin embargo sale al mercado un nuevo receptor, el EM-506.

En la tabla 4.1 se puede encontrar la comparación de los principales parámetros técnicos de los receptores EM-408, iTeed GPS y EM-506 además de un análisis de sus ventajas y desventajas lo que sirvió para realizar la selección definitiva del receptor que se utilizará en el modelo simplificado.

	EM-408	Itead GPS	EM-506
Canales	20	20	48
Sensibilidad	-159 dBm	-159 dBm	-163 dBm
Precisión	10-5 [m]	10 [m]	2.5 [m]
Precio	845	820	520
Micro SD	no	si	no
Antena	Externa/Interna	Externa	Interna
Ventajas	Antena Interna Antena Externa	Interfaz micro SD Antena externa	Mayor cantidad de canales Mayor sensibilidad Mayor precisión Más económico

Desventajas	No incorpora interfaz SD Conector de baja conectividad con antenas GPS	Chipset SiRF III no integrado	No incorpora interfaz SD
-------------	---	-------------------------------	--------------------------

TABLA 4.1 COMPARACIÓN DE PARÁMETROS TÉCNICOS DE RECEPTORES EM-408, ITEAD STUDIO Y EM-506

Observando la tabla de comparación de receptores se decidió tomar el receptor con más canales, mejor sensibilidad y mayor precisión además de ser el más económico por casi 40% menos que los demás receptores y de ser necesaria la interfaz micro SD se tomará esta interfaz del shield Itead Studio.

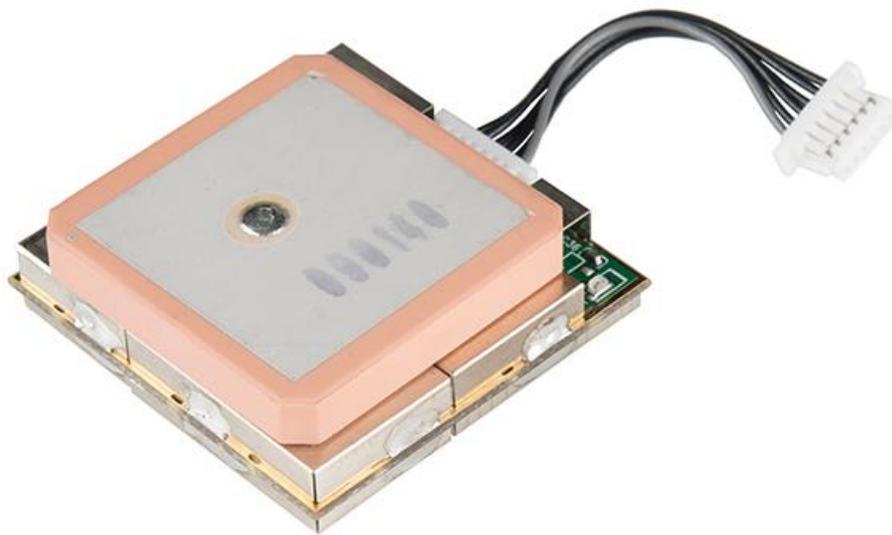


FIGURA 4.6 RECEPTOR EM-506

Interfaz micro SD

Debido a que no se conocen las especificaciones del sistema de monitoreo, nuestro sistema deberá ser configurable con lo que este pueda adaptarse a las condiciones en las que se encuentre el sistema de monitoreo.

Por lo que agregar la interfaz micro SD para almacenar datos podría resultar de utilidad. Si se diera la situación que un elemento del sistema de monitoreo pierda comunicación se perderían los datos recabados. Agregando la interfaz micro SD, se podrían almacenar los datos de posicionamiento para poder recuperarlos en caso de necesitarse para algún análisis o para satisfacer alguna necesidad del sistema de monitoreo.

Entonces, para agregar la interfaz micro SD a nuestro sistema se analizó primeramente la asignación de pines de una tarjeta micro SD mostrada en la figura 4.7



FIGURA 4.7 ASIGNACIÓN DE PINES DE UNA TARJETA MICRO SD

Pin No.	Modo SD			Modo SPI		
	Nombre	Tipo	Descripción	Nombre	Tipo	Descripción
1	DAT2	I/O/PP	Bit 2	RSV		Reservado
2	CD/DAT3	I/O/PP	Card Select/ Bit 3	CS	I	Chip Select
3	CMD	PP	Comando/ Respuesta	DI	I	Data In
4	V _{DD}	S	Alimentación	V _{DD}	S	Alimentación
5	CLK	I	Reloj	SCLK	I	Reloj
6	V _{SS}	S	Tierra	V _{SS}	S	Tierra
7	DAT0	I/O/PP	Bit 0	DO	I/O/PP	Data out
8	DAT1	I/O/PP	Bit 1	RSV		Reservado

S: Fuente de poder I: Entrada O: Salida; PP: Push-Pull

TABLA 4.2 PINES DE TARJETA MICRO SD EN MODO SD Y MODO SPI

Las tarjetas soportan 2 modos de comunicación: Modo SD y Modo SPI. El Modo SD es el modo nativo, y permite mayor velocidad de transferencia que el modo SPI. Las ventajas de este último modo son la simplicidad de conexión con sistemas que utilizan tecnología arduino y la disponibilidad del periférico de comunicaciones SPI en la mayoría de los microcontroladores.

Nosotros usaremos la tarjeta micro SD en modo SPI ya que este modo es compatible con el microcontrolador ATmega328 además, debido al tamaño de la cadena de datos NMEA GPS y a la frecuencia de grabado de datos, no es requerida una velocidad mayor de transmisión y bastará con usar el protocolo SPI.

Conociendo lo anterior se analizó como el iTead Studio GPS shield y el Wireless SD shield incorporan la interfaz SD.

Primero se analizó el iTead Studio GPS shield y de su esquemático se tomaron las partes relevantes a la interfaz micro SD.

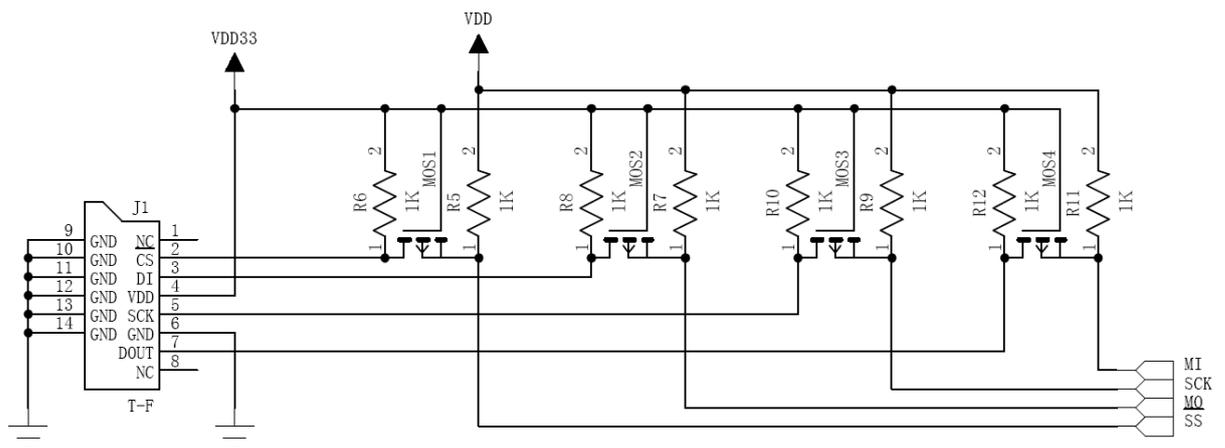


FIGURA 4.8 ESQUEMÁTICO DE INTERFAZ MICRO SD DEL ITEAD STUDIO GPS SHIELD

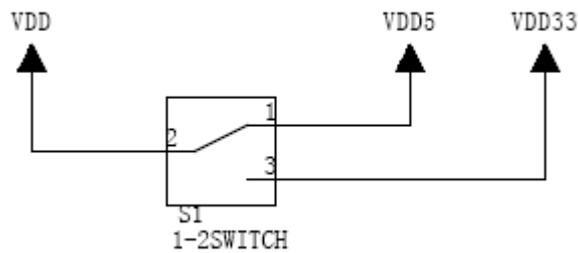


FIGURA 4.9 ESQUEMÁTICO DE SELECCIONADOR DE VOLTAJE DEL ITEAD STUDIO GPS SHIELD

Selector manual de voltaje 5[V] o 3.3[V] dependiendo del receptor GPS con el que trabaje el shield.

Para antena externa VDD=3.3[V]

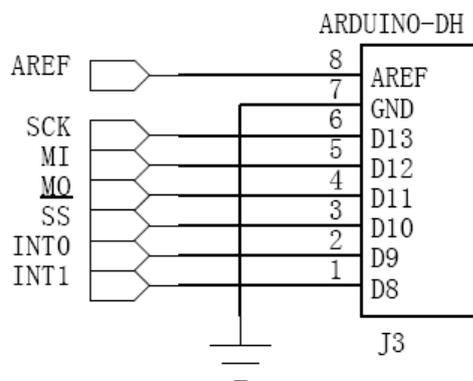


FIGURA 4.10 ESQUEMÁTICO DE CONEXIÓN ENTRE INTERFAZ MICRO SD Y HEADER DE ARDUINO UNO DEL ITEAD STUDIO GPS SHIELD

Los tres componentes del esquemático anterior nos muestran que se necesita un circuito con el cual se controle el voltaje que llegará a los pines SCK, MISO y CS de la tarjeta micro SD ya que el voltaje de operación es de 3.3 [V] y el nivel de voltaje utilizado por el microcontrolador es 5[V].

La conclusión anterior se obtuvo observando y comparando los esquemáticos del Wireless SD Shield y el iTeed Studio GPS Shield ya que ambos incorporan la interfaz micro SD.

Las figuras 4.11, 4.12 y 4.13 muestran los esquemáticos que representan los elementos que otorgan la funcionalidad de almacenamiento de datos al Wireless SD shield.

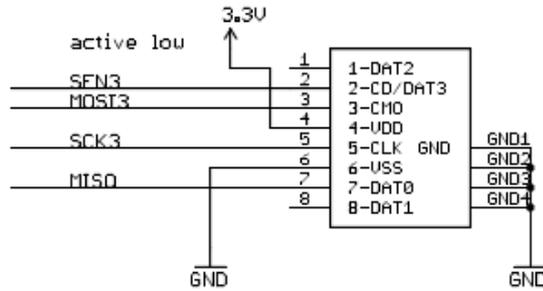


FIGURA 4.11 ESQUEMÁTICO DE CONEXIONES CON TARJETA MICRO SD EN WIRELESS SHIELD

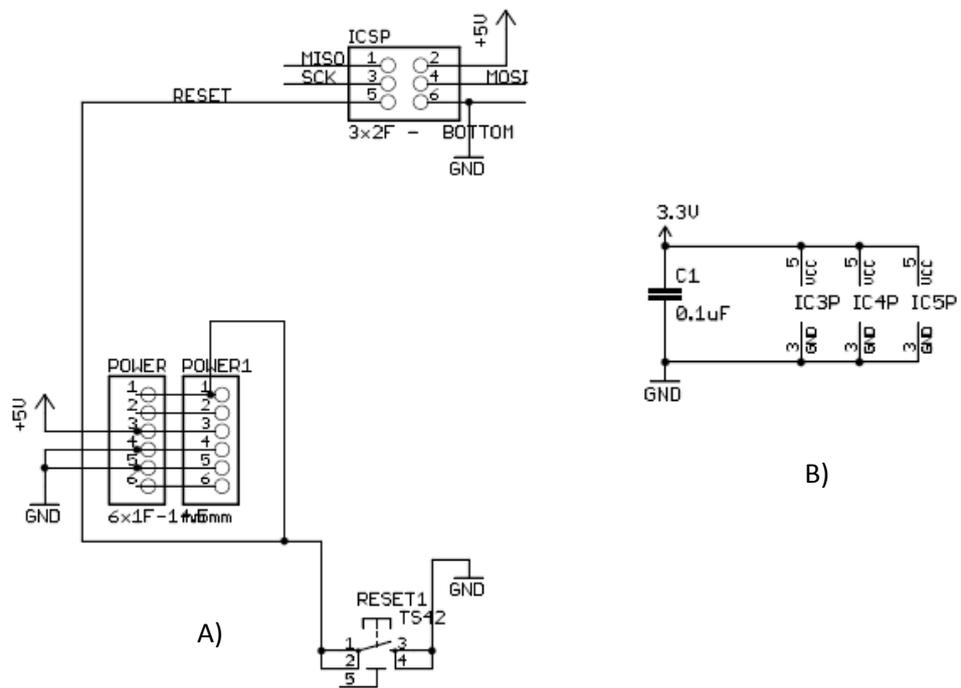


FIGURA 4.12 A) CONEXIONES DE MICRO SD CON ARDUINO EN WIRELESS SHIELD
B) CONEXIONES DE ENERGIZACIÓN DE WIRELESS SHIELD

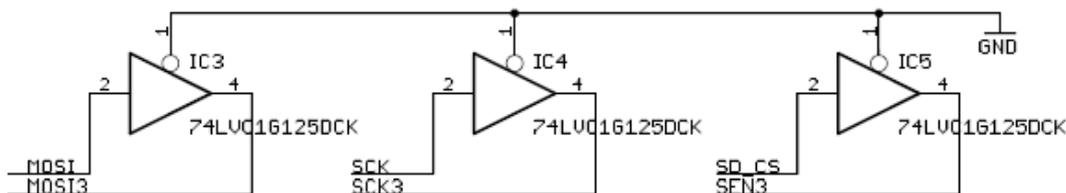


FIGURA 4.13 REGULADOR DE VOLTAJE A 3.3 [V] EN WIRELESS SHIELD

En lugar de los mosfets usados en el iTeed Studio shield este incorpora tres circuitos integrados 74LVC1G125DCK que realizan la misma función que los mosfets, regular el voltaje que llega a los pines SCK, MISO y CS.

Recordando de la distribución de pines de la tarjeta micro SD el pin MISO es un pin de salida que trabajara con 3.3[V] por lo que no tendremos ningún problema al conectarlo directamente al microcontrolador ATmega328.

Por el contrario los pines SCK, MISO y CS de la tarjeta micro SD si deberán ser protegidos ya que son pines de entrada y no queremos que sean dañados con los 5[V] del microcontrolador.

Para nuestro sistema se propone usar un circuito integrado SNLVC125A

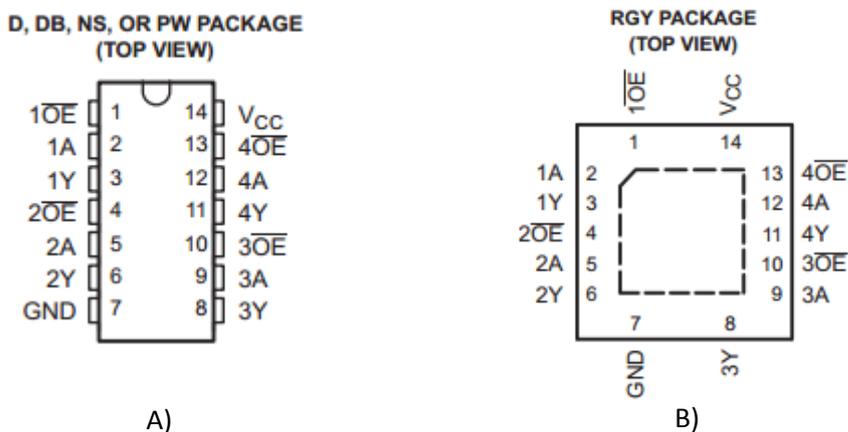


FIGURA 4.14 ASIGNACIÓN DE PINES DE CIRCUITO INTEGRADO SNLVC125A

A) EMPAQUETADO D, DB, NS O PW

B) EMPAQUETADO RGY

Este circuito integrado además de tener pequeñas dimensiones proporciona los tres circuitos controladores de voltaje necesarios para la conexión entre la tarjeta micro SD y el microcontrolador ATmega328.

El SNLVC125A de Texas Instruments está diseñado especialmente para trabajar con un rango de voltaje de 1.65 a 3.6 [V] siendo idóneo para trabajar a 3.3 [V].

A continuación se agrega la tabla lógica del circuito integrado (tabla 4.3) así como su diagrama lógico (figura 4.15) para comprender de una mejor manera su funcionamiento.

Entradas		Salida
OE	A	Y
L	H	H
L	L	L
H	X	Z

TABLA 4.3 TABLA LÓGICA DEL CIRCUITO INTEGRADO SNLVC125A

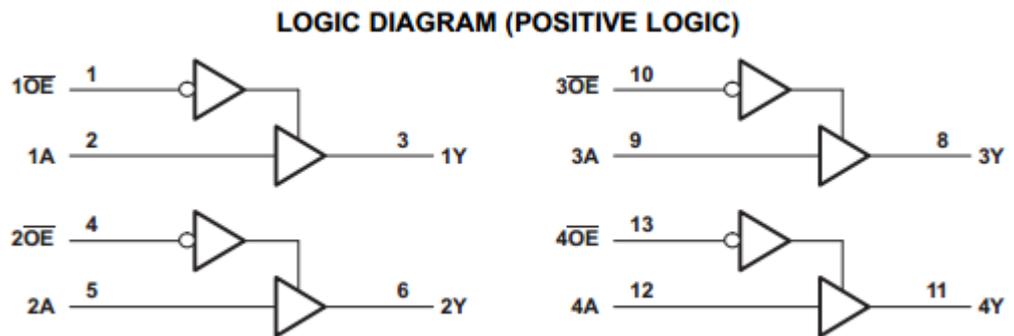


FIGURA 4.15 DIAGRAMA LÓGICO DE CADA BUFFER

Energización del sistema

Considerando todos los componentes a utilizar podemos construir la tabla siguiente:

Componente	Voltaje requerido
Microcontrolador ATmega328	5 [V]
Receptor GPS EM-506	5 [V]
<i>Tarjeta SD</i>	<i>3.3 [V]</i>
<i>Circuito integrado SN74LVC125</i>	<i>3.3 [V]</i>

TABLA 4.4 REQUERIMIENTOS DE POTENCIA DE CADA COMPONENTE DEL SISTEMA SIMPLIFICADO

Con base a eso se propone implementar una alimentación de 9[V] y dos reguladores de voltaje los cuales nos entregaran los voltajes de 5 y 3.3 [V] de una manera estable y sobre todo confiable que además protegerán a nuestros dispositivos de cualquier variación posible.

Sin embargo recordemos que no se conocen los requerimientos del sistema de monitoreo por lo que también se propondrá un sistema en el que no se contará con interfaz micro SD pero que tendrá dimensiones más reducidas.

El sistema simplificado sin interfaz micro SD correspondería a la tabla 4.5

Componente	Voltaje requerido
Microcontrolador ATmega328	5 [V]
Receptor GPS EM-506	5 [V]

TABLA 4.5 REQUERIMIENTOS DE POTENCIA DE CADA COMPONENTE DEL SISTEMA SIMPLIFICADO SIN INTERFAZ MICRO SD

Observando la tabla 4.5 se tiene que únicamente será necesario incorporar un regulador de 5 [V] lo que resulta en una reducción de las dimensiones de la tarjeta PCB al deshacerse del regulador de 3.3 [V] y el circuito integrado SN74LVC125.

Los esquemáticos de conexiones serán mostrados posteriormente en este capítulo.

Diagrama del sistema receptor GPS

En la figura 4.16 se muestra un diagrama que incluye todos los subsistemas del sistema GPS con interfaz micro SD para tener mas en claro el funcionamiento y las conexiones del mismo.

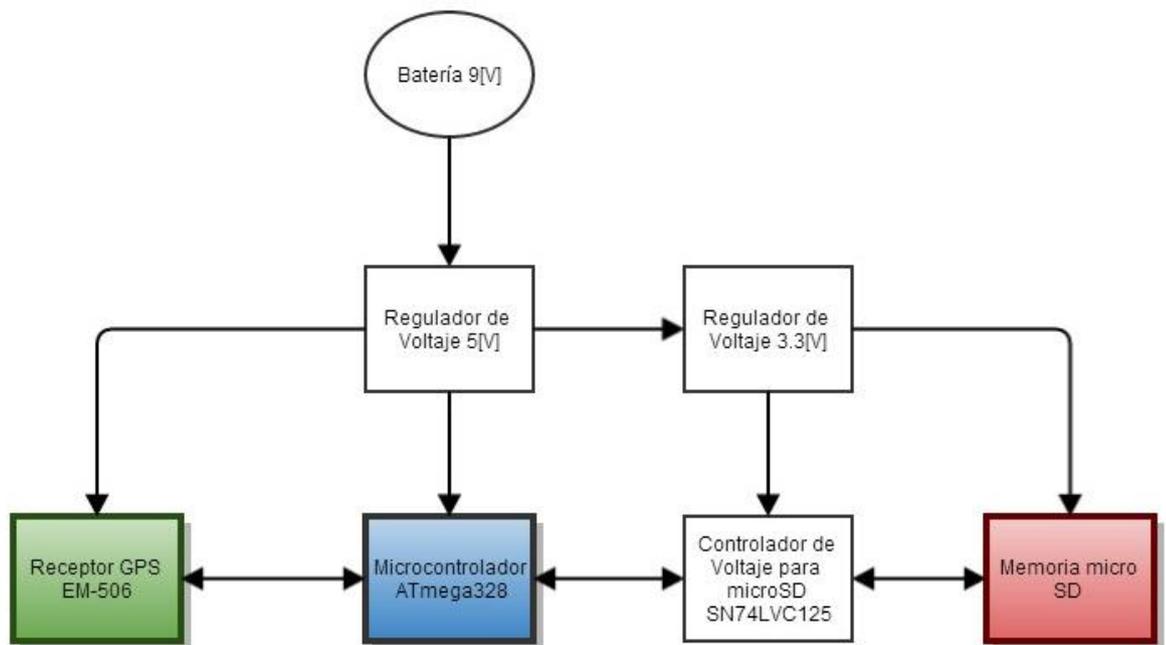


FIGURA 4.16 DIAGRAMA DE BLOQUES DEL SISTEMA GPS CON INTERFAZ MICRO SD

En la figura 4.17 se muestra el diagrama de bloques del sistema simplificado GPS sin interfaz micro SD.

En esta figura se pueden apreciar todos los subsistemas del módulo receptor GPS agrupados en uno solo y se muestra también como se realizaría la conexión con el sistema de monitoreo. Esto es como nuestro sistema entrega los datos de posicionamiento al sistema de monitoreo.

Como se puede notar en la figura 4.17 la conexión entre el sistema de monitoreo y el módulo receptor GPS será mediante el microcontrolador ATmega328 en alguna de sus formas de comunicación mencionadas en el capítulo 2. Es recomendado que esta salida sea de manera serial ya que es la forma de salida que se utiliza en las pruebas (capítulo 5).

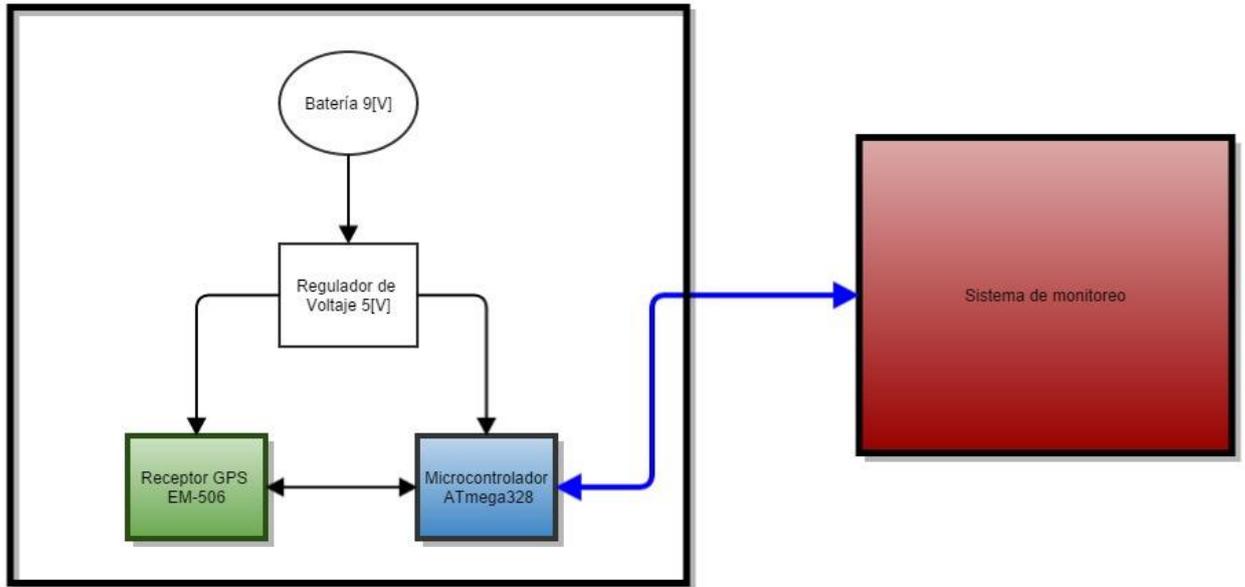


FIGURA 4.17 DIAGRAMA DE BLOQUES DEL SISTEMA SIMPLIFICADO GPS CONECTADO AL SISTEMA DE MONITOREO

Diseño de PCB

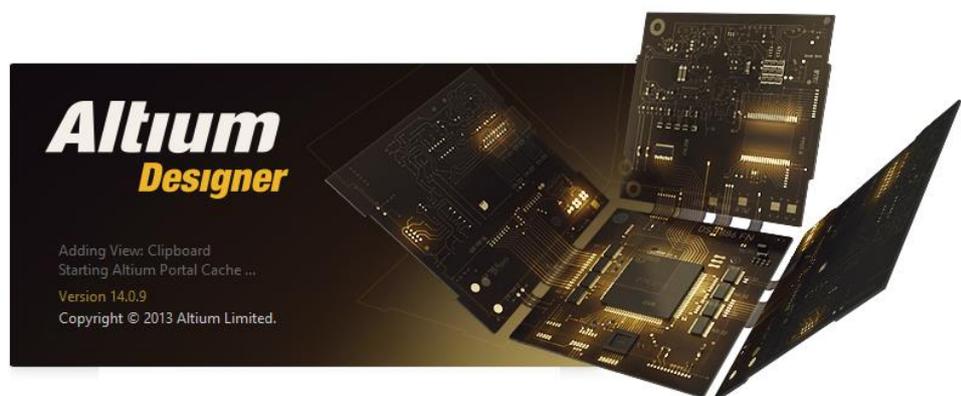


FIGURA 4.18 LOGO ALTIUM DESIGNER 14

Ya contando con todos los componentes del módulo receptor GPS y realizado el diagrama de subsistemas se inició la construcción del prototipo físico. Para esto es necesario utilizar un programa de software profesional el cual nos permita explotar todas las ideas ya desarrolladas en una PCB óptima y funcional.

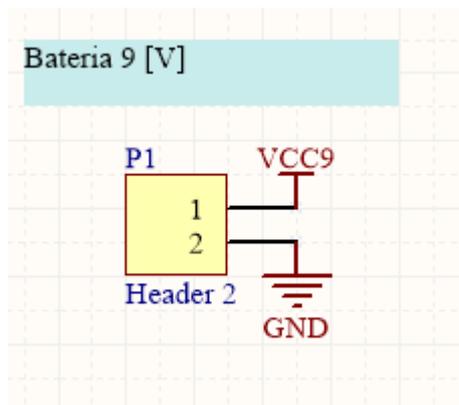
Altium Designer 14 permite realizar lo anterior y además logra producir archivos que son compatibles en la industria para poder fabricar fácilmente la PCB del sistema.

Podemos dividir el trabajo en Altium en dos partes principales:

- Realización de esquemáticos de los componentes
- Diseño de footprints de los componentes y la PCB

Esquemáticos diseñados en Altium Designer 14

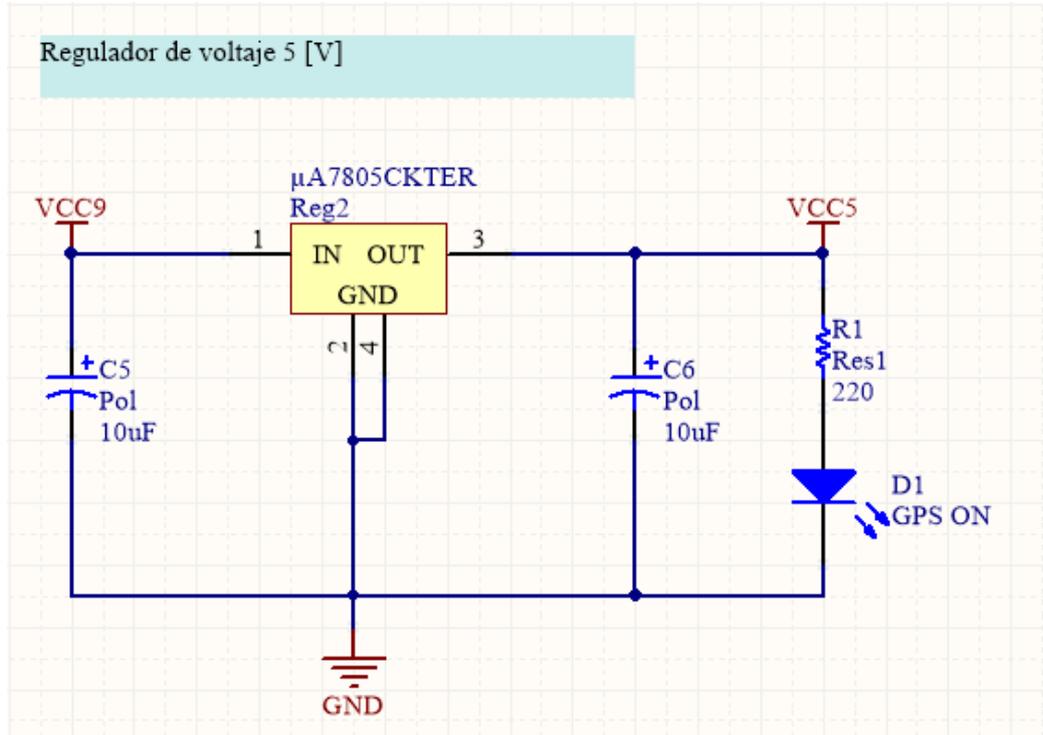
Sección de Potencia



ESQUEMÁTICO 4.1 SUMINISTRO DE VOLTAJE DE 9 [V]

Se alimentará el módulo con un de voltaje de 9 [V] ya que es un valor típico de las baterías que pueden encontrarse en el mercado.

Sección de Potencia

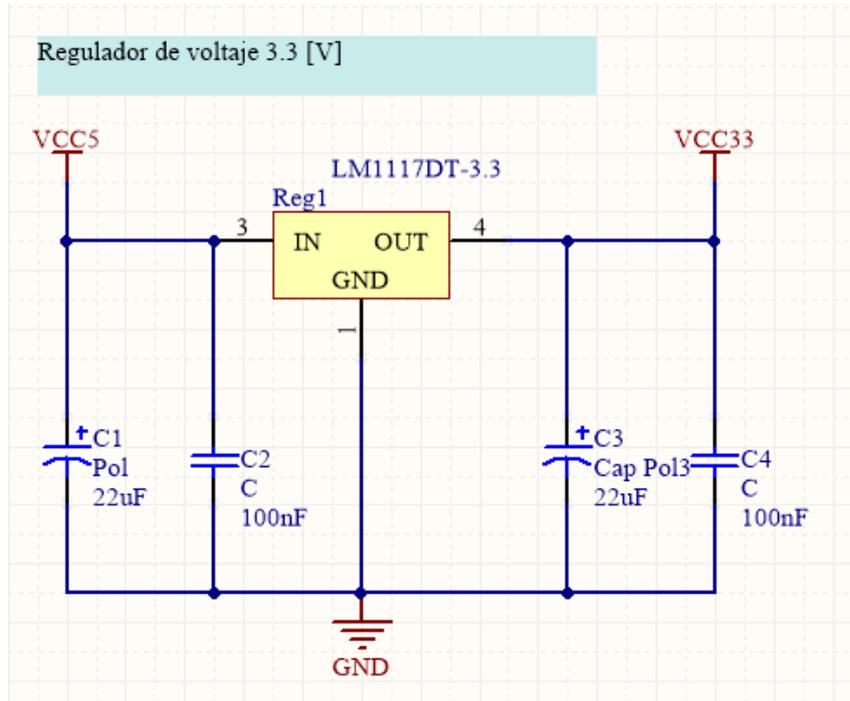


ESQUEMÁTICO 4.2 REGULADOR DE VOLTAJE A 5 [V]

Regulador de voltaje de 9[V] a 5 [V], se incluye un led para indicar que el receptor GPS EM-506 esta encendido. Recordando, este receptor es alimentado con 5[V]. Este regulador también alimentará al microprocesador ATmega328.

El led también nos indicará que el regulador de voltaje a 5 [V] está trabajando y que nuestra fuente de alimentación de 9 [V] está entregando el voltaje necesario.

Sección de Potencia

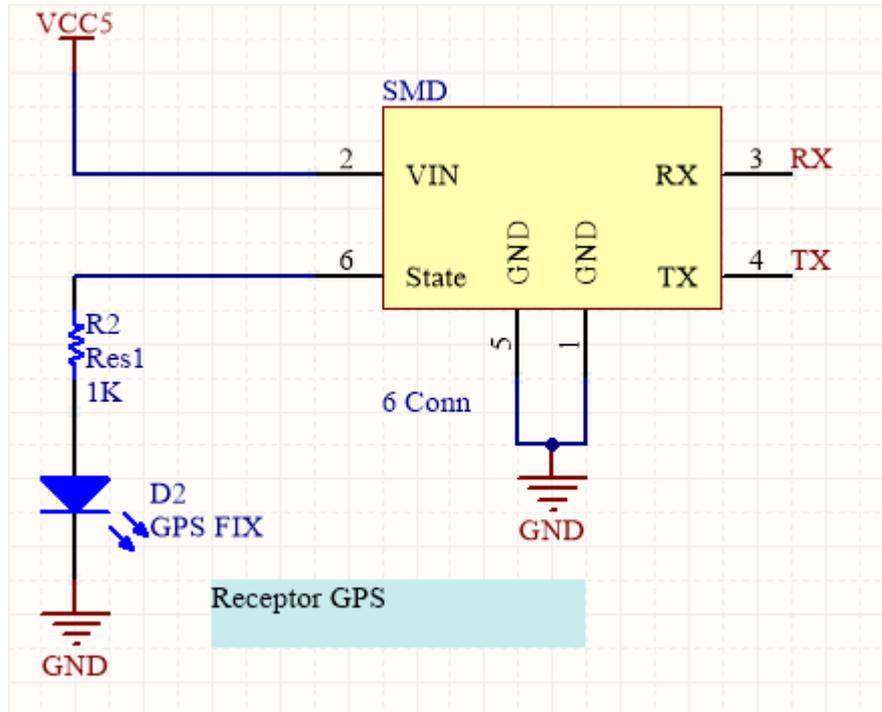


ESQUEMÁTICO 4.3 REGULADOR DE VOLTAJE A 3.3 [V]

Regulador de voltaje de 5[V] a 3.3 [V], con el que alimentaremos la tarjeta micro SD y el circuito integrado SN74LVC125 para hacer el control de voltaje para la comunicación entre el microprocesador y la micro SD.

Este regulador no será necesario para el sistema de dimensiones mínimas ya que no incorpora la interfaz micro SD y como se mencionó anteriormente no se cuenta con ningún elemento que necesite una alimentación de 3 [V].

Receptor GPS

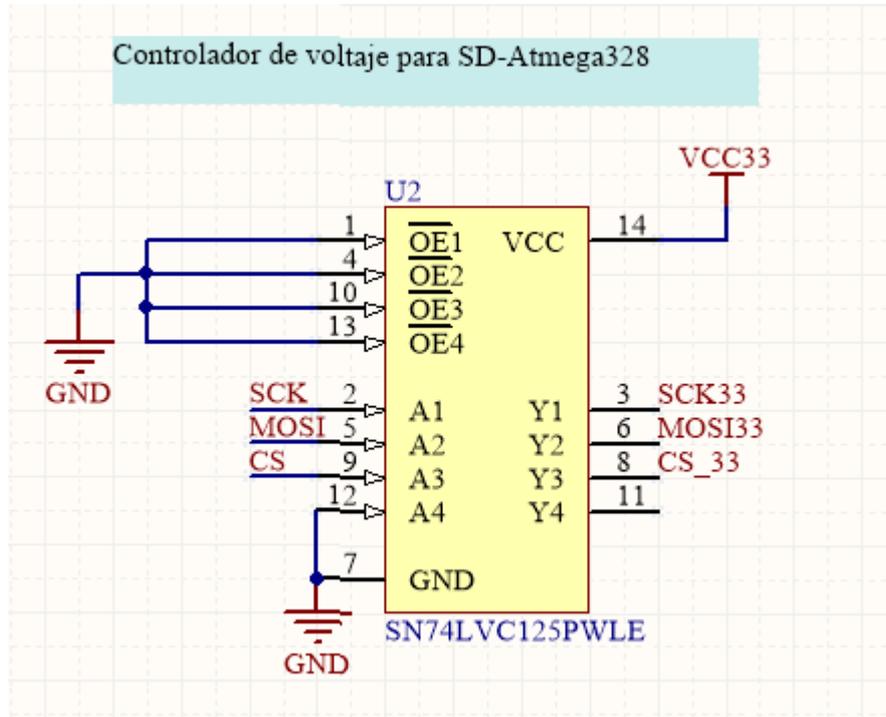


ESQUEMÁTICO 4.4 CONECTOR DEL RECEPTOR GPS

Conector Vertical SMD para conectar el receptor GPS con la PCB mediante los buses RX y TX. Es un conector de tipo JST de 6 conexiones.

Se incluye un led indicador para mostrar cuando se tiene señal fija de recepción GPS.

Controlador de voltaje para comunicación entre SD y ATmega328



ESQUEMÁTICO 4.5 LIMITADOR DE VOLTAJE PARA COMUNICAR TARJETA MICRO SD Y MICROCONTROLADOR ATMEGA328

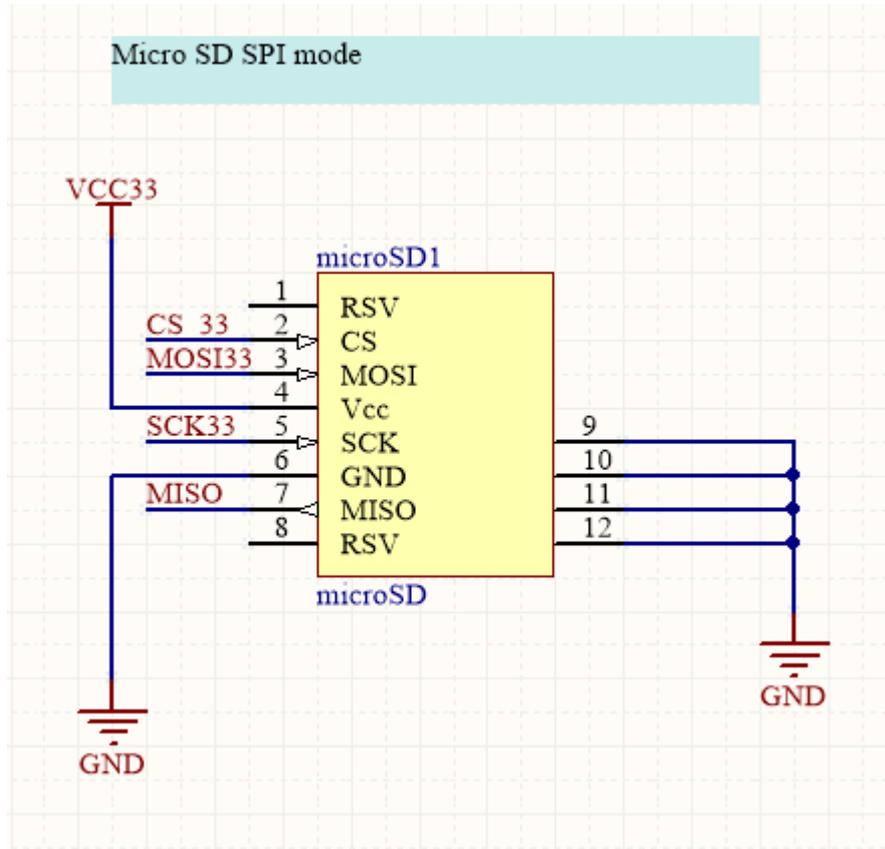
Circuito Integrado de control de voltaje para comunicación entre el microcontrolador con el circuito integrado (buses SCK, MOSI y CS), y la tarjeta micro SD con el circuito integrado (buses SCK33, MOSI33 y CS_33).

Los pines 1, 4, 10 y 13 son conectados a tierra para el funcionamiento deseado del circuito integrado.

El cuarto circuito (pin 12) es conectado a tierra para dejarlo inactivo y que no genere ningún problema a la PCB.

Estas conexiones no serán necesarias en el módulo receptor GPS que no cuenta con interfaz micro SD.

Sujetador de tarjeta micro SD conectada en modo SPI.



ESQUEMÁTICO 4.6 SUJETADOR DE TARJETA MICRO SD CONFIGURADA EN MODO SPI

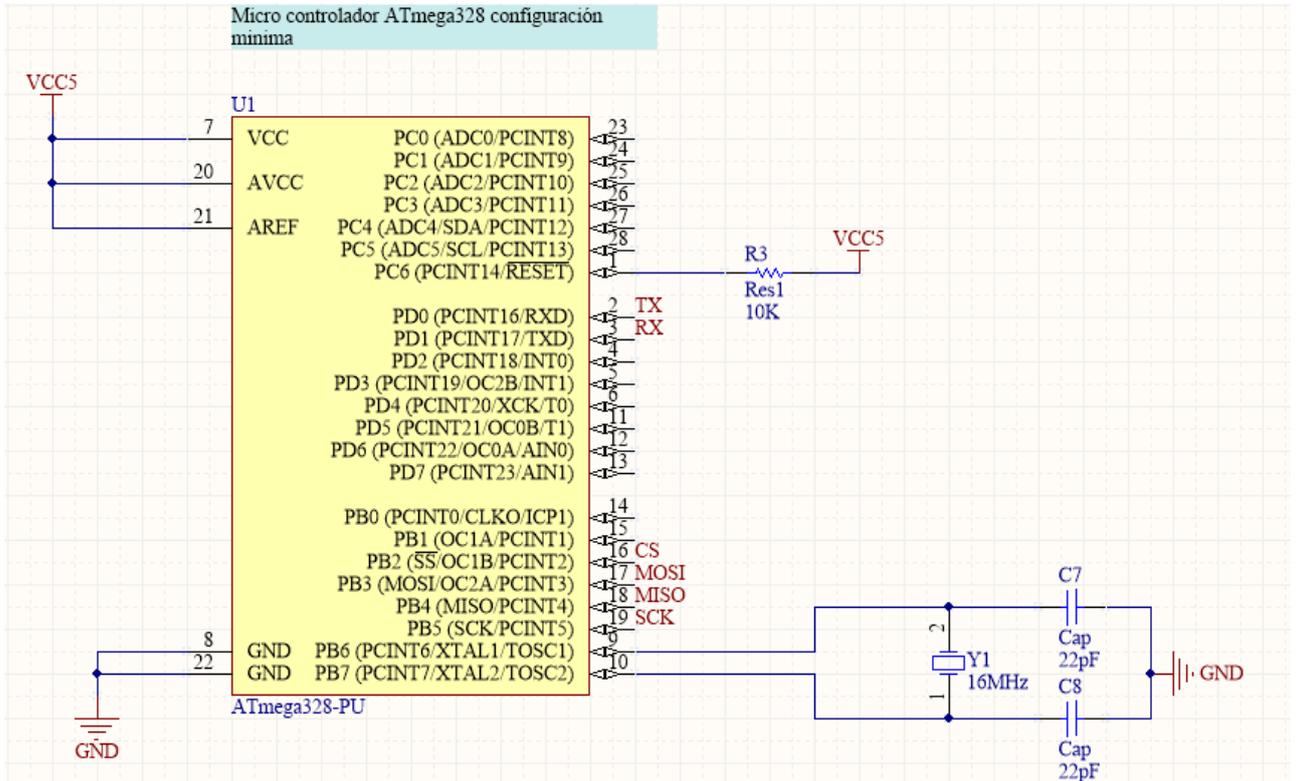
Los pines 1 y 8 no son conectados ya que cuando usamos la tarjeta micro SD en modo SPI estos pines son de uso reservado e inutilizados.

Los pines 9 a 12 representan los puntos a soldar para el sujetador de tarjeta micro SD y estos son usados como tierra.

Los pines 2,3 y 5 son conectados mediante los buses CS_33, MOSI33 y SCK33 al circuito integrado SN74LVC125A.

Estas conexiones no serán necesarias en el módulo receptor GPS que no cuenta con interfaz micro SD.

Microcontrolador ATmega328 con su configuración mínima



ESQUEMÁTICO 4.7 MICROCONTROLADOR ATMEGA328 CON SU CONFIGURACIÓN MÍNIMA

Microcontrolador ATmega328 con la configuración mínima propuesta anteriormente además de estar conectado al receptor GPS EM-506 mediante los buses TX y RX, y tener las conexiones necesarias con la tarjeta micro SD mediante los buses (CS, MOSI, MISO, SCK).

Para módulo receptor GPS que no cuenta con interfaz micro SD únicamente se retiraron las conexiones de los buses CS, MOSI, MISO y SCK.

Los demás pines no serán conectados ya que como se describe en las especificaciones del microcontrolador ATmega328, no afectarán el funcionamiento del mismo.

Archivos “GERBER” para producción de PCB

Para el diseño de los archivos gerber se tomó el módulo sin interfaz micro SD ya que las dimensiones de la PCB se lograron reducir considerablemente respecto a la que si incorpora la interfaz micro SD y esto resulta en la posibilidad de integrar nuestro sistema al sistema de monitoreo de una forma mucho más sencilla.

Altium designer permite hacer múltiples configuraciones para la PCB dependiendo de las necesidades del proyecto.

Para este diseño se contempló pistas en una cara de la tarjeta ya que la producción de tarjetas con pistas en más de una cara es más costoso.

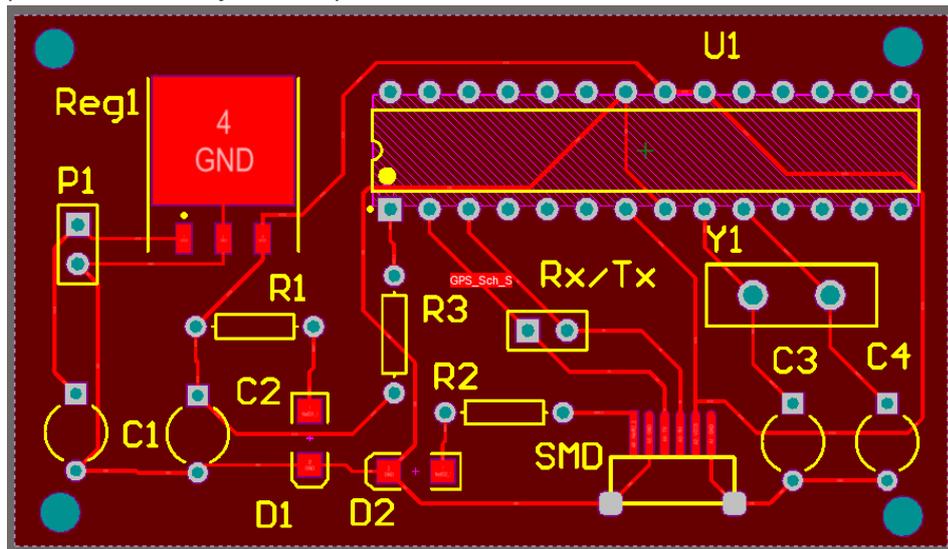


FIGURA 4.20 DISEÑO PROPUESTO DE PCB, MÓDULO RECEPTOR GPS

En la tabla 4.6 se listan los componentes del módulo receptor GPS con los elementos de la PCB.

Componente	Nomenclatura en PCB
Regulador 5 [V]	Reg1
Atmega328	U1
Conector vertical SMA para Receptor GPS 506	SMD
Cristal de 16 [MHz]	Y1
Capacitor de 22 [pF]	C3, C4
Capacitor de 10 [uF]	C1, C2
Resistencia de 10 [KΩ]	R3
Resistencia de 220 [KΩ]	R1

Resistencia de 1 [KΩ]	R2
Diodo LED	D1, D2
Puntos de conexión para obtener datos del módulo GPS	RX/TX

TABLA 4.6 CORRESPONDENCIA DE LOS ELEMENTOS DEL MÓDULO GPS CON LOS ELEMENTOS SITUADOS EN LA PCB

Si se diera el caso en que la naturaleza y requerimientos del sistema de monitoreo nos indican que la interfaz micro SD es requerida sin importar que las dimensiones del módulo receptor aumenten, se podrá agregar el gerber del sujetador micro SD presentado en la figura 4.21 satisfaciendo ambas situaciones.

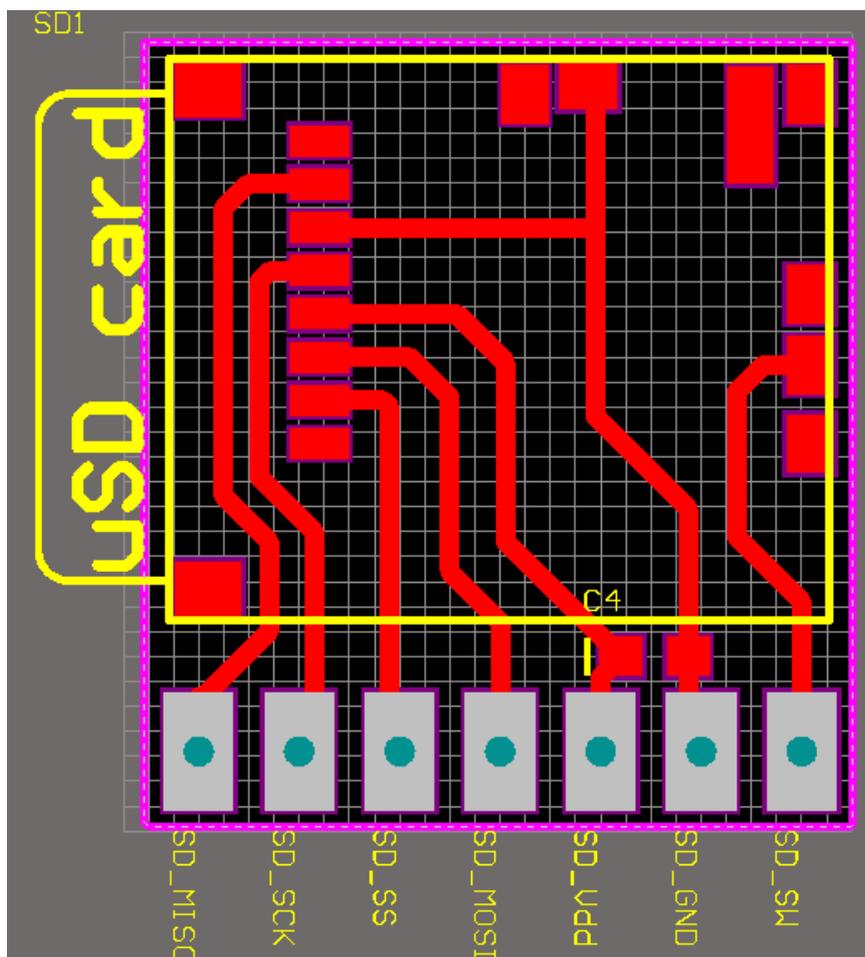


FIGURA 4.21 ARCHIVO GERBER DE SUJETADOR DE TARJETA MICRO SD

Para producir una PCB en la industria son necesarios los archivos GERBER conformados por los archivos CAMtastic.

Las figuras 4.22, 4.23, 4.24 y 4.25 son algunos de los archivos GERBER que pueden producirse.

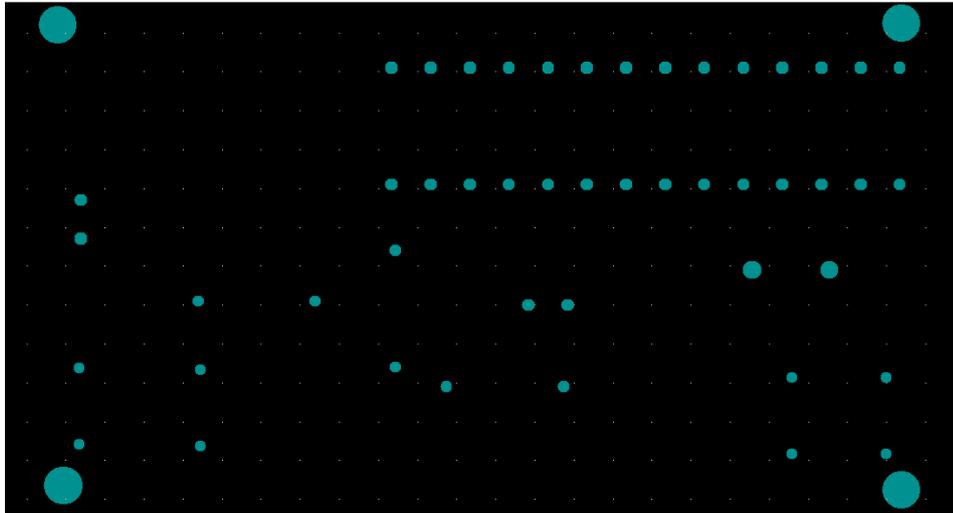


FIGURA 4.22 ARCHIVO CAMTASTIC .DRILL

La figura 4.22 muestra todos los hoyos que se necesitan realizar en la PCB.

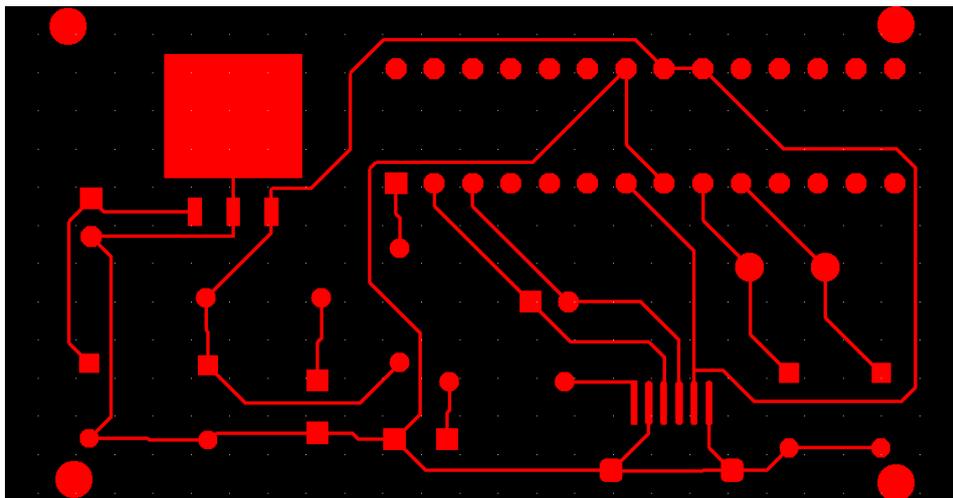


FIGURA 4.23 ARCHIVO CAMTASTIC .GTL

La figura 4.23 muestra todas las pistas de la capa superior de la PCB.

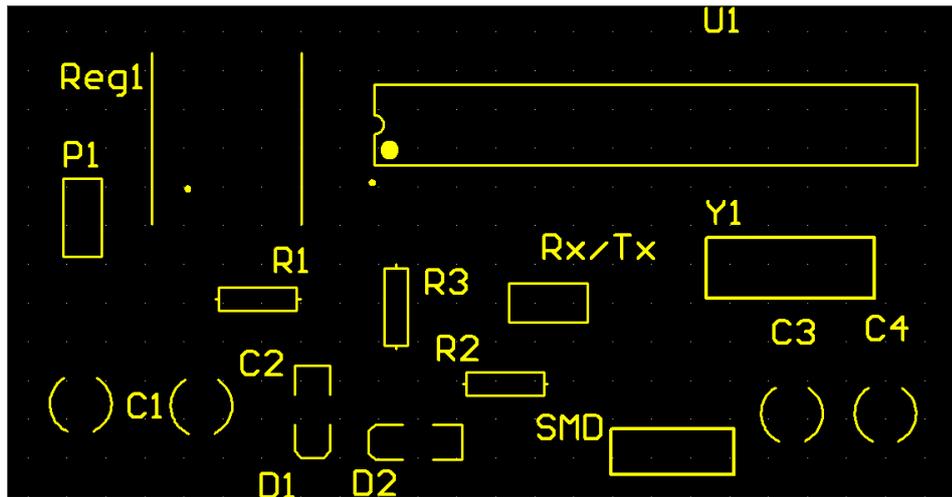


FIGURA 4.24 ARCHIVO CAMTASTIC .GTO

La figura 4.24 muestra todas las etiquetas de la capa superior de la PCB

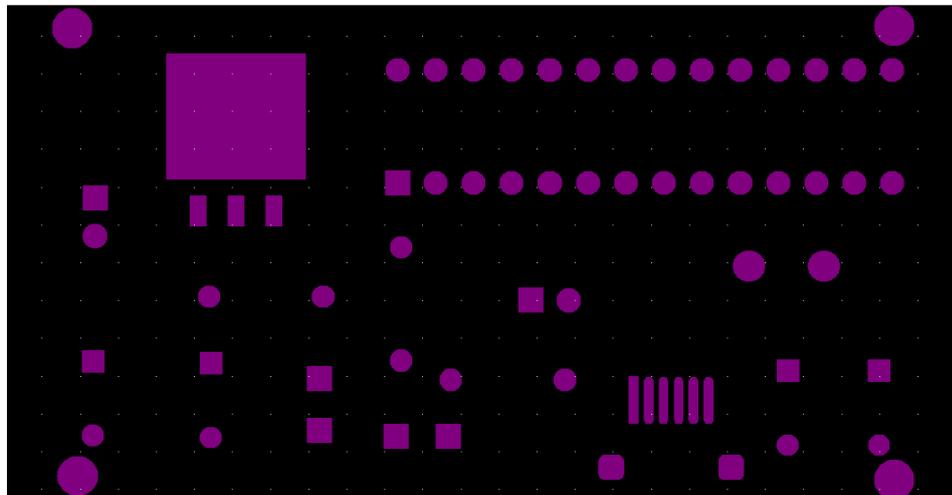


FIGURA 4.25 ARCHIVO CAMTASTIC .GTS

La figura 4.25 muestra todos los componentes de la capa superior de soldado de la PCB

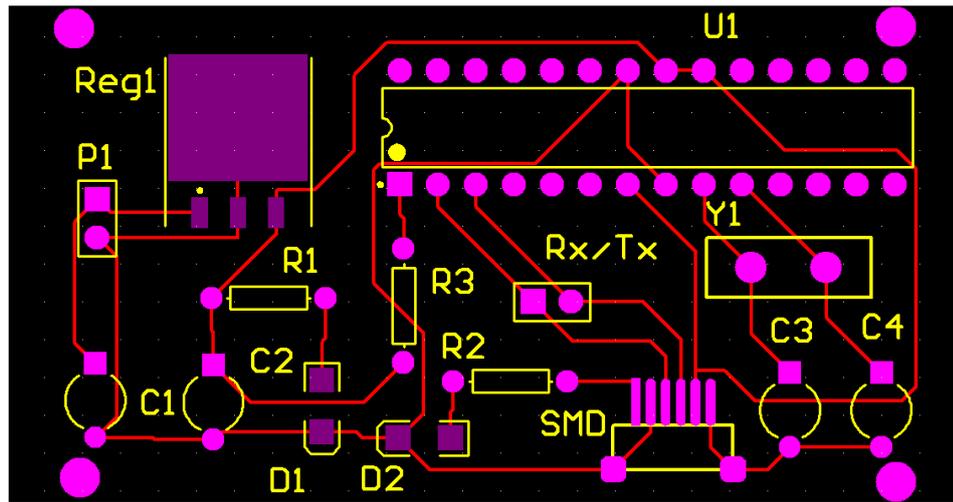


FIGURA 4.26 ARCHIVO CAMTASTIC .CAM

La figura 4.26 muestra todas las capas que se mandaron a imprimir de la PCB incluyendo capa mecánica, capa de etiquetas, capa de soldado, capa de pistas y hoyos necesarios en la PCB.

Las dimensiones físicas de la PCB diseñada se pueden apreciar en la tabla 4.7

Dimensiones Físicas de la PCB	
Alto	3.4 [cm]
Largo	5.9 [cm]

TABLA 4.7 DIMENSIONES FÍSICAS DE LA PCB DEL MÓDULO GPS

Capítulo 5 Pruebas y Resultados

Durante el desarrollo del proyecto se realizaron diversas pruebas a cada uno de los subsistemas de manera independiente. Estas pruebas se llevaron a cabo como parte del diseño del sistema, con la finalidad de evaluar el desempeño de cada elemento en particular y para garantizar una buena integración de los subsistemas. La realización de estas pruebas fue crucial para seleccionar los elementos que constituyen el módulo receptor GPS.

En este capítulo se presentan las pruebas realizadas con los sistemas receptores descritos en el capítulo 3 Sistemas Receptores GPS.

Además, se presenta un prototipo con el que se validará el diseño de la PCB del módulo receptor GPS presentada en el capítulo 4 Integración del Sistema Simplificado.

Protocolo de Validación

Para la validación de los resultados que se presentarán en este capítulo se utilizaron dos métodos:

- 1) Comparación de datos obtenidos con los sistemas experimentales y prototipo con datos obtenidos con receptor BU353 de GlobalSat y su programa de software GPSINFO.
- 2) Comparación de datos obtenidos con los sistemas experimentales y prototipo con datos generados con Google Maps y Pixelis Maps, programas de uso libre, disponibles en Internet.

El receptor BU353 es un receptor de alta sensibilidad, bajo consumo de potencia y ultra compacto. Cuenta con conexión USB haciéndolo compatible con casi cualquier dispositivo personal. Este receptor cuenta con un chipset SiRF IV permitiendo proveer al usuario de un mejor desempeño incluso en condiciones de baja visibilidad superando las características de los receptores utilizados a lo largo de este proyecto, los que incorporan un chipset SiRF III.

El dispositivo receptor BU353 cuenta con un imán integrado en su superficie haciendo su colocación mucho más sencilla para el usuario.

Para la interpretación de los datos recibidos por el receptor BU353 se utilizó el software GPSINFO ya que es el programa que proporciona GlobalSat para hacer uso del receptor BU353. Dicho programa permite interpretar las cadenas de datos NMEA recibidas por el receptor mostrando información como es fecha, tiempo, velocidad, latitud, longitud, estado entre otros.

El programa muestra los satélites con los que se tiene conexión. La pantalla de este programa y sus características se pueden apreciar en la figura 5.1.

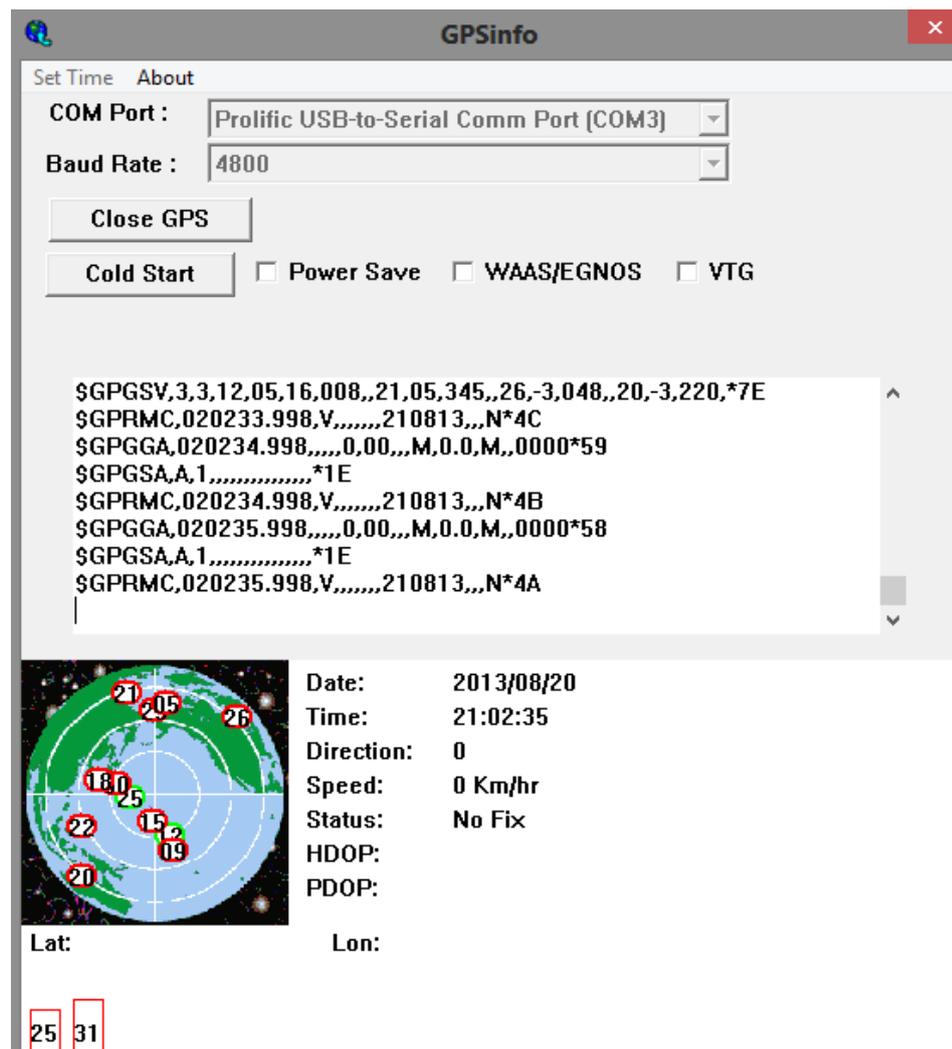


FIGURA 5.1 OBTENCIÓN DE DATOS DE RECEPTOR BU353 CON GPSINFO

Las posiciones obtenidas gracias a este receptor se pudieron comparar con las obtenidas con los sistemas receptores GPS y con el prototipo del módulo GPS validando los resultados de ambos.

El receptor BU353 se puede observar en la figura 5.2



FIGURA 5.2 RECEPTOR BU353 DE GLOBALSAT

La segunda manera de validar los datos que se fueron obteniendo con los sistemas experimentales y el prototipo, fue posible gracias a Google Maps y Pixelis Maps ambos programas disponibles en internet.

Sus respectivas direcciones son las siguientes:

- <http://www.maps.pixelis.es/>
- <https://www.google.com.mx/maps/@23.5541314,-102.6205,5z>

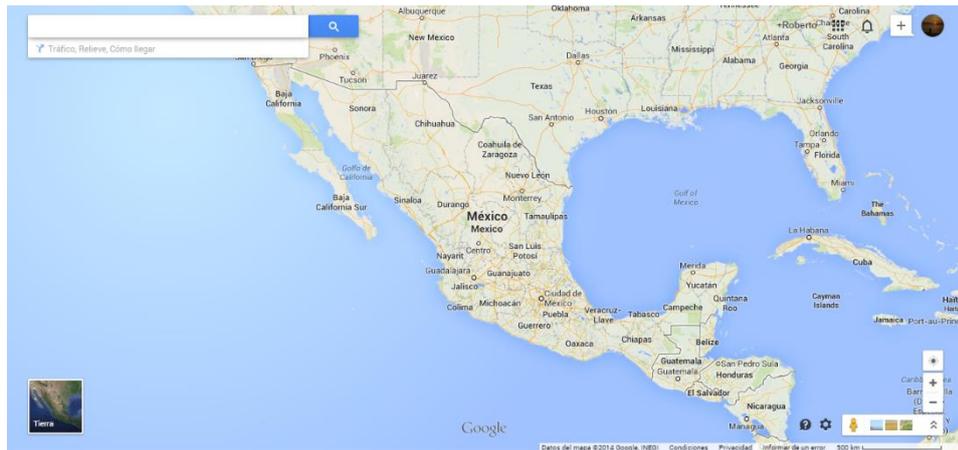


FIGURA 5.3 PÁGINA DE INICIO DE GOOGLE.MX

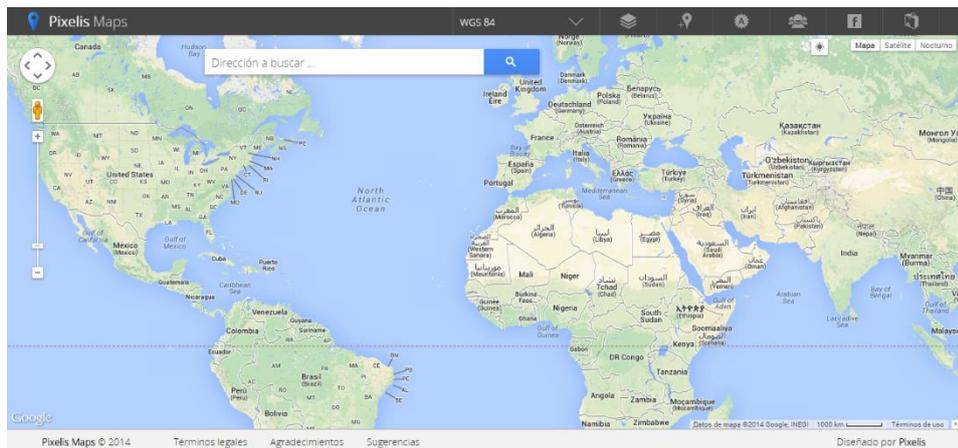


FIGURA 5.4 PÁGINA DE INICIO DE PIXELIS MAPS

Para corroborar los datos obtenidos en el proyecto, se siguió el siguiente procedimiento:

- 1) Obtener medidas validas de latitud y longitud mediante alguno de los receptores utilizados a lo largo del proyecto o con el prototipo implementado.
- 2) Ingresar la latitud y longitud en el buscador de Pixelis Maps y esperar hasta que se genere un mapa con la ubicación ingresada
- 3) Ir a Google Maps y buscar la dirección de donde se realizaron las medidas y generar un mapa
- 4) Comparar las posiciones que generaron los mapas de Pixelis Maps y Google Maps

- 5) En caso de tener posiciones iguales o con una tolerancia no mayor a la precisión indicada en las hojas de especificaciones de cada receptor se podrá concluir que se tiene una medida correcta.

Receptor EM-408

Teniendo el sistema ya ensamblado como se describió en el capítulo 3, se cargó a la tarjeta arduino UNO el programa GPS_simple_test_EM408.ino disponible en la sección Anexos.

Este programa permite:

- Corroborar el correcto funcionamiento del receptor.
- Obtener Latitud, Longitud y número de satélites.
- Visualizar de manera serial número de caracteres, cadenas y errores recibidos.
- Visualizar de ser necesario todos los caracteres recibidos.
- Conocer la precisión con la que se realizan las mediciones

Los resultados obtenidos con buen clima y línea de vista se aprecian en la figura 5.5

```
CHARS=5931 SENTENCES=0 CSUM ERR=1
CHARS=6175 SENTENCES=0 CSUM ERR=1
LAT=19.266510 LON=-99.121192 SAT=3 PREC=460 CHARS=6576 SENTENCES=3 CSUM ERR=1
LAT=19.268249 LON=-99.122009 SAT=3 PREC=460 CHARS=6868 SENTENCES=5 CSUM ERR=1
LAT=19.269189 LON=-99.122459 SAT=3 PREC=460 CHARS=7179 SENTENCES=7 CSUM ERR=1
LAT=19.269779 LON=-99.122749 SAT=3 PREC=460 CHARS=7484 SENTENCES=9 CSUM ERR=1
```

FIGURA 5.5 MONITOR SERIAL CON DATOS RECIBIDOS POR EM-408, BUEN CLIMA Y LÍNEA DE VISTA

Los resultados bajo condiciones de lluvia y escasa línea de vista se aprecian en la figura 5.6

```
CHARS=15881 SENTENCES=0 CSUM ERR=1
CHARS=15969 SENTENCES=0 CSUM ERR=1
CHARS=16276 SENTENCES=0 CSUM ERR=1
LAT=19.240200 LON=-99.145431 SAT=3 PREC=2140 CHARS=16656 SENTENCES=1 CSUM ERR=1
LAT=19.240200 LON=-99.145431 SAT=3 PREC=2140 CHARS=16777 SENTENCES=2 CSUM ERR=1
```

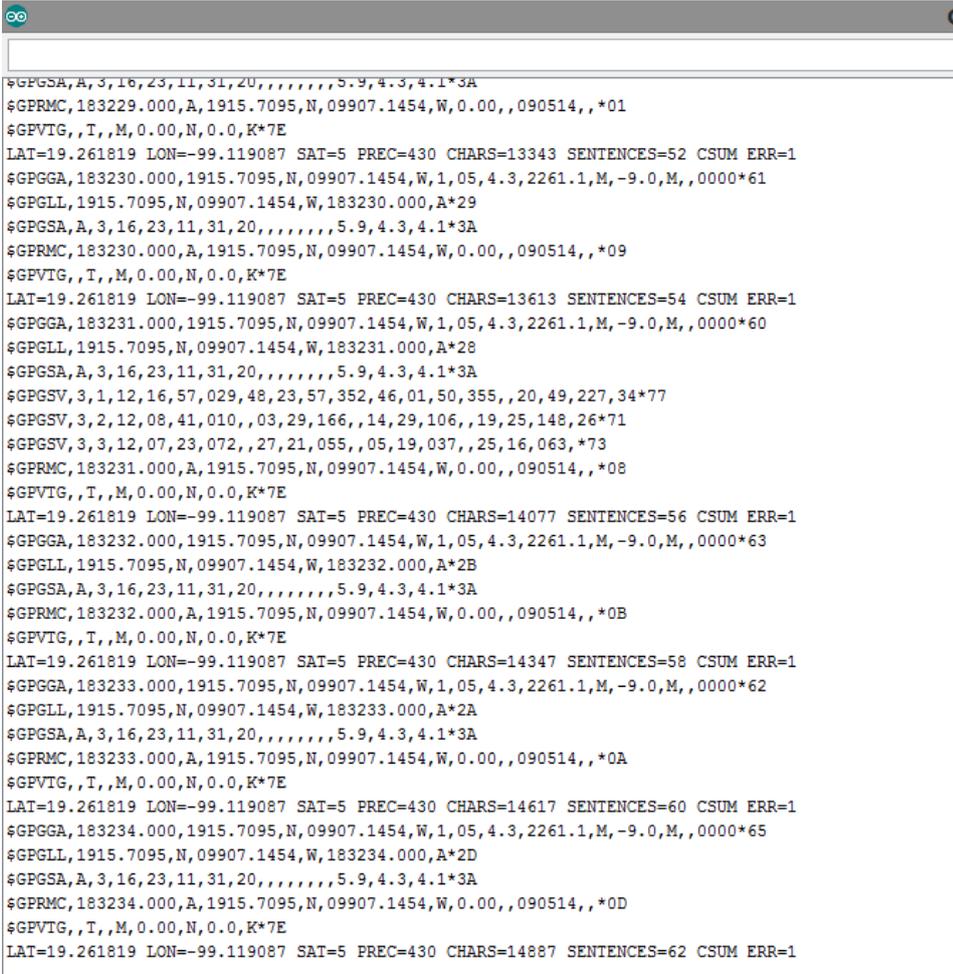
FIGURA 5.6 MONITOR SERIAL CON DATOS RECIBIDOS POR EM-408, LLUVIA Densa Y LÍNEA DE VISTA

Si comparamos los resultados mostrados en las figuras 5.5 y 5.6 podemos observar que la visibilidad que se tiene así como las condiciones climáticas afectan en desempeño del receptor.

En línea de vista y con buen clima el receptor únicamente necesito recibir 6,175 caracteres (20 segundos) antes de determinar la posición.

Con escasa línea de vista y lluvia abundante el receptor requirió recibir 16,276 caracteres (1 minuto) antes de determinar la posición.

Cuando se quita el comentario “Serial.write(c);” del programa GPS_simple_test_EM408.ino se observa un monitor serial como el de la figura 5.7



```

$GPGSA,A,3,16,23,11,31,20,,,,,,,,,5.9,4.3,4.1*3A
$GPRMC,183229.000,A,1915.7095,N,09907.1454,W,0.00,,090514,,*01
$GPGVTG,,T,,M,0.00,N,0.0,K*7E
LAT=19.261819 LON=-99.119087 SAT=5 PREC=430 CHARS=13343 SENTENCES=52 CSUM ERR=1
$GPGGA,183230.000,1915.7095,N,09907.1454,W,1,05,4.3,2261.1,M,-9.0,M,,0000*61
$GPGGL,1915.7095,N,09907.1454,W,183230.000,A*29
$GPGSA,A,3,16,23,11,31,20,,,,,,,,,5.9,4.3,4.1*3A
$GPRMC,183230.000,A,1915.7095,N,09907.1454,W,0.00,,090514,,*09
$GPGVTG,,T,,M,0.00,N,0.0,K*7E
LAT=19.261819 LON=-99.119087 SAT=5 PREC=430 CHARS=13613 SENTENCES=54 CSUM ERR=1
$GPGGA,183231.000,1915.7095,N,09907.1454,W,1,05,4.3,2261.1,M,-9.0,M,,0000*60
$GPGGL,1915.7095,N,09907.1454,W,183231.000,A*28
$GPGSA,A,3,16,23,11,31,20,,,,,,,,,5.9,4.3,4.1*3A
$GPGSV,3,1,12,16,57,029,48,23,57,352,46,01,50,355,,20,49,227,34*77
$GPGSV,3,2,12,08,41,010,,03,29,166,,14,29,106,,19,25,148,26*71
$GPGSV,3,3,12,07,23,072,,27,21,055,,05,19,037,,25,16,063,*73
$GPRMC,183231.000,A,1915.7095,N,09907.1454,W,0.00,,090514,,*08
$GPGVTG,,T,,M,0.00,N,0.0,K*7E
LAT=19.261819 LON=-99.119087 SAT=5 PREC=430 CHARS=14077 SENTENCES=56 CSUM ERR=1
$GPGGA,183232.000,1915.7095,N,09907.1454,W,1,05,4.3,2261.1,M,-9.0,M,,0000*63
$GPGGL,1915.7095,N,09907.1454,W,183232.000,A*2B
$GPGSA,A,3,16,23,11,31,20,,,,,,,,,5.9,4.3,4.1*3A
$GPRMC,183232.000,A,1915.7095,N,09907.1454,W,0.00,,090514,,*0B
$GPGVTG,,T,,M,0.00,N,0.0,K*7E
LAT=19.261819 LON=-99.119087 SAT=5 PREC=430 CHARS=14347 SENTENCES=58 CSUM ERR=1
$GPGGA,183233.000,1915.7095,N,09907.1454,W,1,05,4.3,2261.1,M,-9.0,M,,0000*62
$GPGGL,1915.7095,N,09907.1454,W,183233.000,A*2A
$GPGSA,A,3,16,23,11,31,20,,,,,,,,,5.9,4.3,4.1*3A
$GPRMC,183233.000,A,1915.7095,N,09907.1454,W,0.00,,090514,,*0A
$GPGVTG,,T,,M,0.00,N,0.0,K*7E
LAT=19.261819 LON=-99.119087 SAT=5 PREC=430 CHARS=14617 SENTENCES=60 CSUM ERR=1
$GPGGA,183234.000,1915.7095,N,09907.1454,W,1,05,4.3,2261.1,M,-9.0,M,,0000*65
$GPGGL,1915.7095,N,09907.1454,W,183234.000,A*2D
$GPGSA,A,3,16,23,11,31,20,,,,,,,,,5.9,4.3,4.1*3A
$GPRMC,183234.000,A,1915.7095,N,09907.1454,W,0.00,,090514,,*0D
$GPGVTG,,T,,M,0.00,N,0.0,K*7E
LAT=19.261819 LON=-99.119087 SAT=5 PREC=430 CHARS=14887 SENTENCES=62 CSUM ERR=1

```

FIGURA 5.7 MONITOR SERIAL CON TODAS LAS CADENAS DE DATOS RECIBIDAS POR EM-408

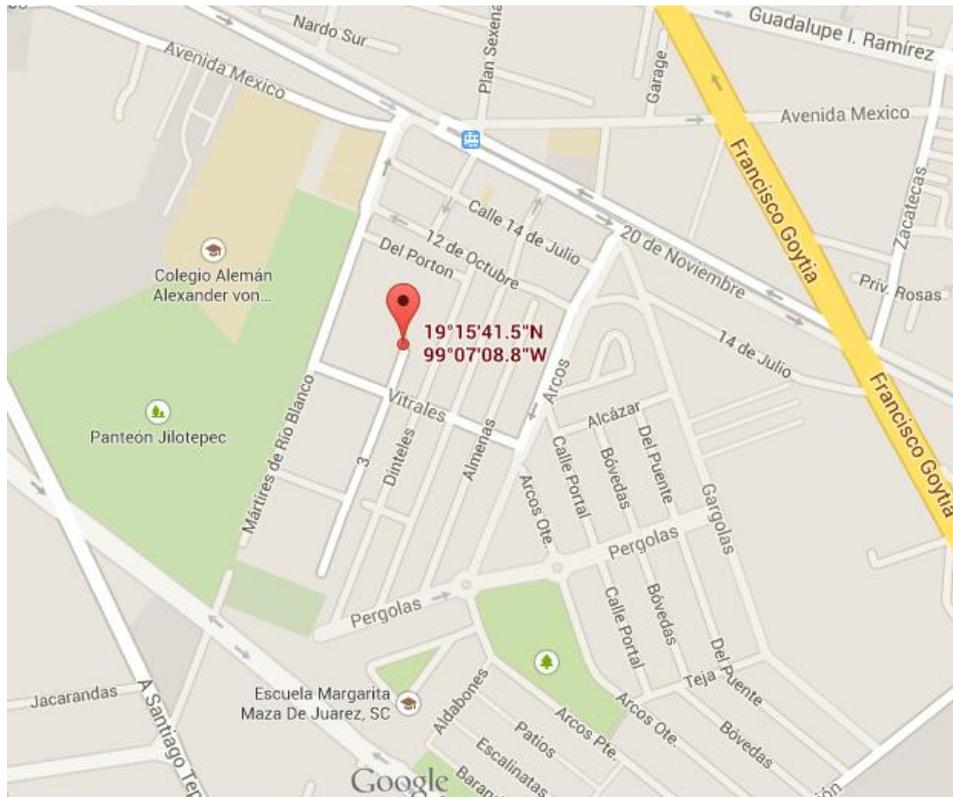


FIGURA 5.9 MAPA GENERADO CON GOOGLE MAPS

La diferencia que se observa entre el mapa de la figura 5.8 y el mapa de la figura 5.9 es de aproximadamente 8 [m] por lo que concluimos que efectivamente es una medición válida.

Receptor EM-408 con interfaz micro SD y arduino Mega

Para probar el funcionamiento de la interfaz micro SD se utilizó la configuración descrita en el capítulo 3 Sistema con módulo receptor EM-408, arduino MEGA y compatibilidad micro SD. Además se cargó en el microcontrolador de la tarjeta arduino MEGA el programa GPS_logger_mega.ino disponible en la sección Anexos.

Este sistema permite:

- Verificar la versión de librería TinyGPS con la que se cuenta

- Mostrar serialmente los siguientes parámetros: latitud, longitud, número de satélites, fecha, tiempo, altitud y curso.
- Corroborar el correcto funcionamiento del receptor.
- Visualizar de manera serial número de caracteres, cadenas y errores recibidos.
- Visualizar de ser necesario todos los caracteres recibidos.
- Generar archivos con extensión .txt o .csv
- Guardar o sobrescribir archivos con extensión .txt o .csv en una tarjeta micro SD
- Identificar el origen de una posible falla en el sistema

Cuando se inicializa el monitor serial se puede observar una pantalla como la presentada en la figura 5.10. En esta se aprecia que aún no se tiene determinada la posición debido a que el receptor requiere cierto tiempo para determinar la posición, sin embargo se puede ver que se están recibiendo caracteres puesto que en la columna de caracteres recibidos el número acumulado de estos va aumentando con cada segundo.

```
TinyGPS library v. 12
Roberto Ortega GPS DATA LOGGER

Sizeof(gpsobject) = 115

Sats HDOP Latitude Longitude Fix Date Time Date Alt Course Speed Card Distance Course Card Chars Sentences Checksum
(deg) (deg) Age Age (m) --- from GPS ---- ---- to London ---- RX RX Fail
-----
**** **** ***** ***** **** ***** ***** **** ***** ***** ***** **** 0 0.00 *** 223 0 0
invalid,***** ,*****
**** **** ***** ***** **** ***** ***** **** ***** ***** ***** **** 0 0.00 *** 598 0 0
invalid,***** ,*****
**** **** ***** ***** **** ***** ***** **** ***** ***** ***** **** 0 0.00 *** 777 0 0
invalid,***** ,*****
**** **** ***** ***** **** ***** ***** **** ***** ***** ***** **** 0 0.00 *** 971 0 0
invalid,***** ,*****
```

FIGURA 5.10 MONITOR SERIAL CON DATOS RECIBIDOS POR ARDUINO MEGA Y RECEPTOR EM-408

En caso de tener alguna falla con la tarjeta micro SD o si no se cuenta con una insertada correctamente en el sujetador se desplegará una pantalla como la mostrada en la figura 5.11

```

SD card no encontrada
**** ** 0 0.00 *** 257 0

No se pudo acceder al archivo CSV
**** ** 0 0.00 *** 585 0

No se pudo acceder al archivo CSV
**** ** 0 0.00 *** 752 0

```

FIGURA 5.11 MONITOR SERIAL CON DATOS RECIBIDOS POR ARDUINO MEGA Y RECEPTOR EM-408 SIN TARJETA MICRO SD

En este caso también se puede ver que se están recibiendo caracteres puesto que en la columna de caracteres recibidos el número acumulado de estos va aumentando con cada segundo.

Lo anterior resulta muy útil para poder determinar el origen de una falla en caso de presentarse ya que el programa detecta si el problema está relacionado con la tarjeta micro SD o si no se reciben caracteres indicando entonces que el problema es originado en el receptor.

En la figura 5.12 se muestra el monitor obtenido con un sistema en el que se tiene bien insertada la tarjeta micro SD pero que no se conectó el receptor.

```

TinyGPS library v. 12
Roberto Ortega GPS DATA LOGGER

sizeof(gpsobject) = 115

Sats HDOP Latitude Longitude Fix Date Time Date Alt Course Speed Card Distance Course Card Chars Sentences
(deg) (deg) Age --- from GPS --- ---- to London ---- RX RX
-----
**** ** 0 0.00 *** 0 0
invalid,***** ,*****
**** ** 0 0.00 *** 0 0
invalid,***** ,*****
**** ** 0 0.00 *** 0 0
invalid,***** ,*****
**** ** 0 0.00 *** 0 0
invalid,***** ,*****
**** ** 0 0.00 *** 0 0
invalid,***** ,*****
**** ** 0 0.00 *** 0 0
invalid,***** ,*****
**** ** 0 0.00 *** 0 0
invalid,***** ,*****

```

FIGURA 5.12 MONITOR SERIAL CON DATOS RECIBIDOS POR ARDUINO MEGA, TARJETA MICRO SD Y RECEPTOR DESCONECTADO

En el caso de la figura 5.12 no está indicado ningún problema relacionado con la tarjeta micro SD y tampoco se tiene determinada la posición pero a diferencia de los casos de las figuras 5.10 y 5.11, el número de caracteres recibidos está en cero lo que nos indica que algún problema se tiene con el receptor siendo que este no está conectado.

Además de un monitor como los mostrados en las figuras 5.10, 5.11 y 5.12 el programa también genera un archivo con extensión .csv o .txt dependiendo de la extensión que se coloque en el programa .ino

En este caso se mostrarán algunas pruebas realizadas generando archivos .csv que es una extensión compatible con Excel

Para esta prueba primero se realizaron mediciones en movimiento con el sistema y se recuperó la tarjeta micro SD para manejar los datos recabados por el receptor y almacenados por la tarjeta SD en una computadora portátil.

El sistema genera el archivo LOG.csv que puede ser abierto con ayuda de Excel. En este se aprecia una tabla de tres columnas como la de la figura 5.13

	A	B	C
1	09/10/2013 21:18	19.26792	-99.10676
2	09/10/2013 21:18	19.26797	-99.10674
3	09/10/2013 21:18	19.268	-99.10678
4	09/10/2013 21:18	19.26798	-99.10682
5	09/10/2013 21:18	19.26795	-99.10686
6	09/10/2013 21:18	19.26796	-99.10691
7	09/10/2013 21:18	19.26799	-99.10697
8	09/10/2013 21:18	19.26798	-99.10694
9	09/10/2013 21:18	19.26797	-99.10692
10	09/10/2013 21:18	19.2679	-99.10685
11	09/10/2013 21:18	19.26792	-99.1069
12	09/10/2013 21:18	19.26793	-99.10691
13	09/10/2013 21:18	19.26795	-99.1069
14	09/10/2013 21:18	19.26796	-99.10688
15	09/10/2013 21:18	19.26798	-99.10685
16	09/10/2013 21:18	19.26799	-99.10682
17	09/10/2013 21:18	19.26804	-99.10673
18	09/10/2013 21:18	19.26808	-99.10665
19	09/10/2013 21:18	19.26807	-99.10667
20	09/10/2013 21:18	19.26805	-99.1067
21	09/10/2013 21:18	19.26804	-99.10673
22	09/10/2013 21:18	19.26803	-99.10674
23	09/10/2013 21:18	19.26803	-99.10674

FIGURA 5.13 ARCHIVO LOG.CSV GENERADO POR SISTEMA RECEPTOR CON EM-408 Y ARDUINO MEGA

La tabla de la figura 5.13 contiene tres columnas, la primera contiene la hora y fecha del receptor, la segunda la Latitud y por último la tercer columna contiene la Longitud.

El siguiente paso fue colocar los encabezados de cada columna como se muestra en la figura 5.14 y volver a guardar el archivo.

	A	B	C
1	Date/Time	Latitude	Longitude
2	09/10/2013 21:18	19.26792	-99.10676
3	09/10/2013 21:18	19.26797	-99.10674
4	09/10/2013 21:18	19.268	-99.10678
5	09/10/2013 21:18	19.26798	-99.10682
6	09/10/2013 21:18	19.26795	-99.10686
7	09/10/2013 21:18	19.26796	-99.10691
8	09/10/2013 21:18	19.26799	-99.10697
9	09/10/2013 21:18	19.26798	-99.10694
10	09/10/2013 21:18	19.26797	-99.10692

FIGURA 5.14 ARCHIVO LOG.CSV GENERADO POR SISTEMA RECEPTOR CON EM-408 Y ARDUINO MEGA CON ENCABEZADOS

Ya con los encabezados se puede dar diferentes aplicaciones a los datos recabados gracias al programa de software de uso libre GPS Visualizer disponible en Internet en la siguiente página:

- <http://www.gpsvisualizer.com/>

GPS Visualizer es una herramienta en línea que genera mapas a partir de datos y coordenadas geográficas.

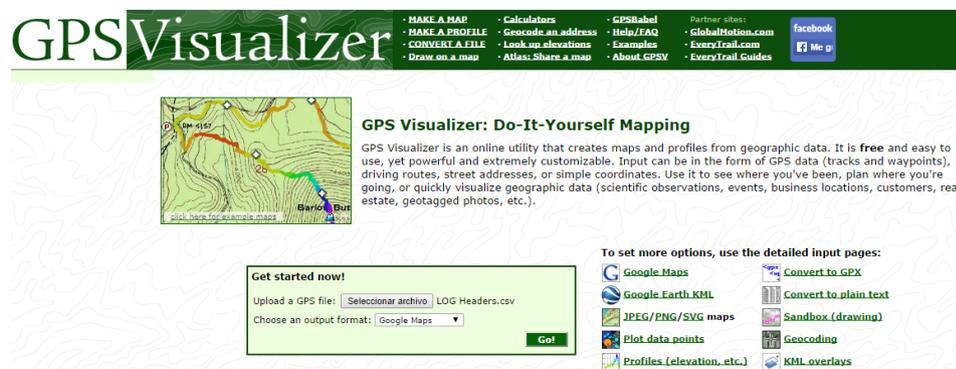


FIGURA 5.15 PANTALLA DE INICIO GPS VISUALIZER

GPS ofrece diferentes formas de salida como lo son:

- Imágenes con extensión JPEG, PNG o SVG
- Mapa en Google maps
- Mapa en Google Earth extensión KML

Utilizando GPS Visualizer se generaron los mapas de dos diferentes recorridos realizados con el sistema receptor.

Con el primer recorrido se generó un mapa con extensión KML el cual, se pudo visualizar en Google Earth. Este mapa se aprecia en la figura 5.16 y el recorrido descrito se observa marcado en línea roja.

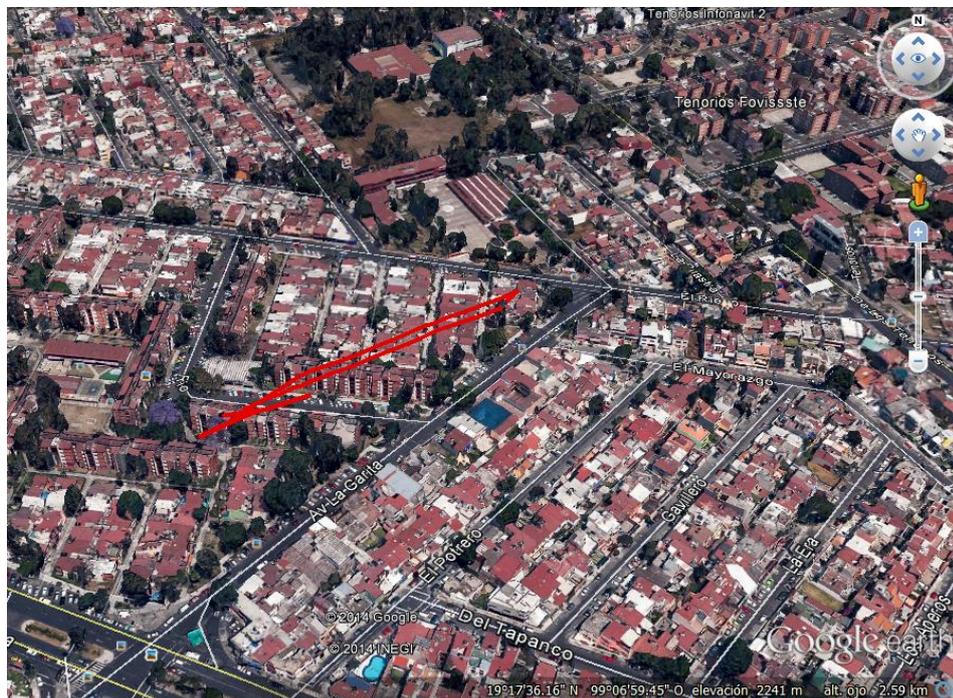


FIGURA 5.16 MAPA EXTENSIÓN KML GENERADO CON GPS VISUALIZER

Con el segundo recorrido se generó un Mapa en Google Maps que se puede apreciar en la figura 5.17, el recorrido descrito se observa mediante los puntos rojos en el mapa.

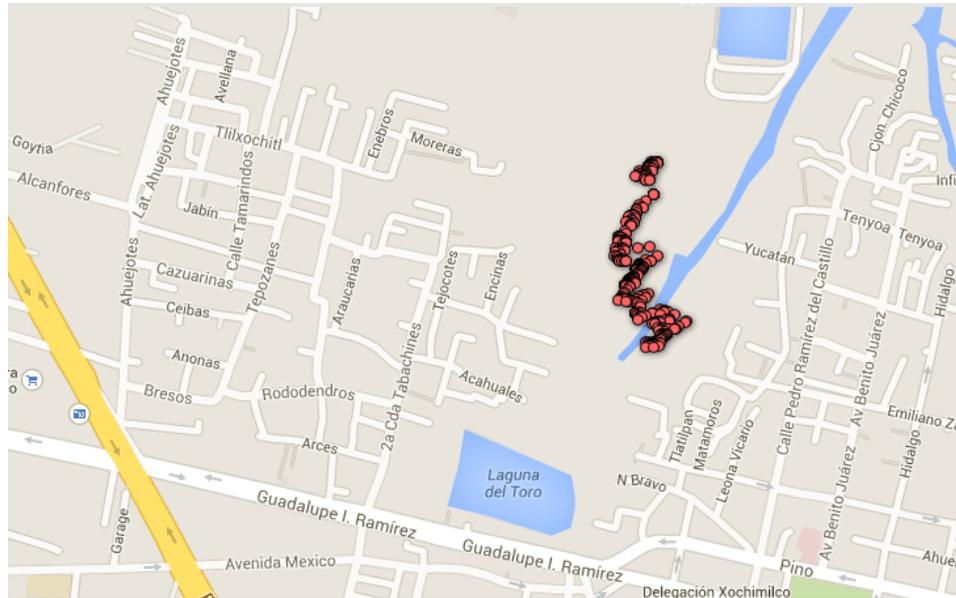


FIGURA 5.17 MAPA EN GOOGLE MAPS GENERADO CON GPS VISUALIZER

Receptor iTead Studio

Teniendo el sistema ya ensamblado como se describió en el capítulo 3, se cargó a la tarjeta arduino UNO el programa GPS_iTead_parameters.ino disponible en la sección Anexos.

Este programa permite:

- Corroborar el correcto funcionamiento del receptor.
- Obtener Latitud, Longitud, tiempo, velocidad en [Kmph] y [mph] así como la altitud en [m].
- Visualizar de manera serial número de caracteres.
- Visualizar de ser necesario todas las cadenas recibidas.

Cuando se inicializa el monitor serial se puede observar una pantalla como la presentada en la figura 5.18

```
CHARS=4787
CHARS=4954
CHARS=5121
CHARS=5288
LAT=19.261800 LON=-99.118988 Time=1200425.0000 SPEED kmph=0.0000 SPEED mph=-1 ALTITUD meters=2267.600097 CHARS=5586
LAT=19.261819 LON=-99.119049 Time=1200525.0000 SPEED kmph=3.4077 SPEED mph=2 ALTITUD meters=2254.000000 CHARS=6040
LAT=19.261760 LON=-99.119003 Time=1200600.0000 SPEED kmph=1.2038 SPEED mph=0 ALTITUD meters=2255.100097 CHARS=6320
LAT=19.261770 LON=-99.119010 Time=1200700.0000 SPEED kmph=0.0000 SPEED mph=0 ALTITUD meters=2255.899902 CHARS=6588
LAT=19.261770 LON=-99.119010 Time=1200800.0000 SPEED kmph=0.0000 SPEED mph=0 ALTITUD meters=2255.899902 CHARS=6856
LAT=19.261770 LON=-99.119010 Time=1200900.0000 SPEED kmph=0.0000 SPEED mph=0 ALTITUD meters=2255.899902 CHARS=7318
LAT=19.261770 LON=-99.119010 Time=1201000.0000 SPEED kmph=0.0000 SPEED mph=0 ALTITUD meters=2255.899902 CHARS=7588
LAT=19.261770 LON=-99.119010 Time=1201100.0000 SPEED kmph=0.0000 SPEED mph=0 ALTITUD meters=2255.899902 CHARS=7856
```

FIGURA 5.18 MONITOR SERIAL CON DATOS RECIBIDOS POR ITEAD SHIELD

Ya que se trata de una prueba estática no se tiene variación de velocidad. En la sexta línea del monitor serial de la figura 5.18 se tiene una variación de velocidad de 3 kilómetros por hora. Esto es debido a que las primeras mediciones normalmente se realizan con pocos satélites en línea de vista y eso hace que podamos tener ligeras variaciones en la posición.

El receptor interpreta estas ligeras variaciones como movimiento. Después de adquirir más satélites en línea de vista, el receptor deja de tener estas variaciones y como se ve en la figura 5.18 a partir de la octava línea ya no se registra ninguna variación de velocidad.

El mapa generado a partir de Pixelis Maps se muestra en la figura 5.19

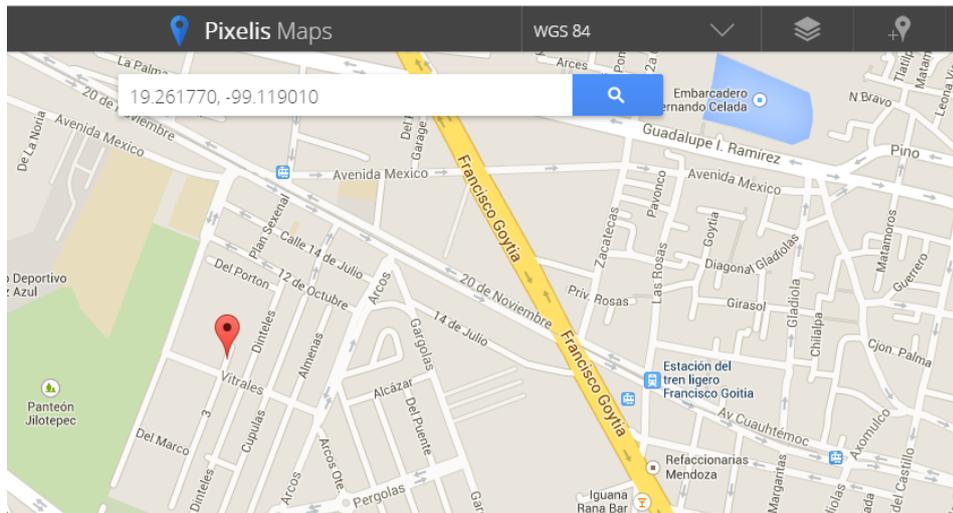


FIGURA 5.19 MAPA GENERADO CON PIXELIS MAPS A PARTIR DE DATOS RECABADOS CON RECEPTOR ITEAD STUDIO

A partir del mapa de la figura 5.19 se puede saber que el receptor está funcionando de manera correcta ya que generó un resultado prácticamente idéntico al resultado generado con el receptor EM-408.

Si no se contara con este método de validación se tendría que recurrir a comparar las cadenas NMEA recibidas.

Para esta prueba tomaremos la cadena \$GPGA del receptor BU353 y la compararemos con la que genera nuestro sistema receptor.

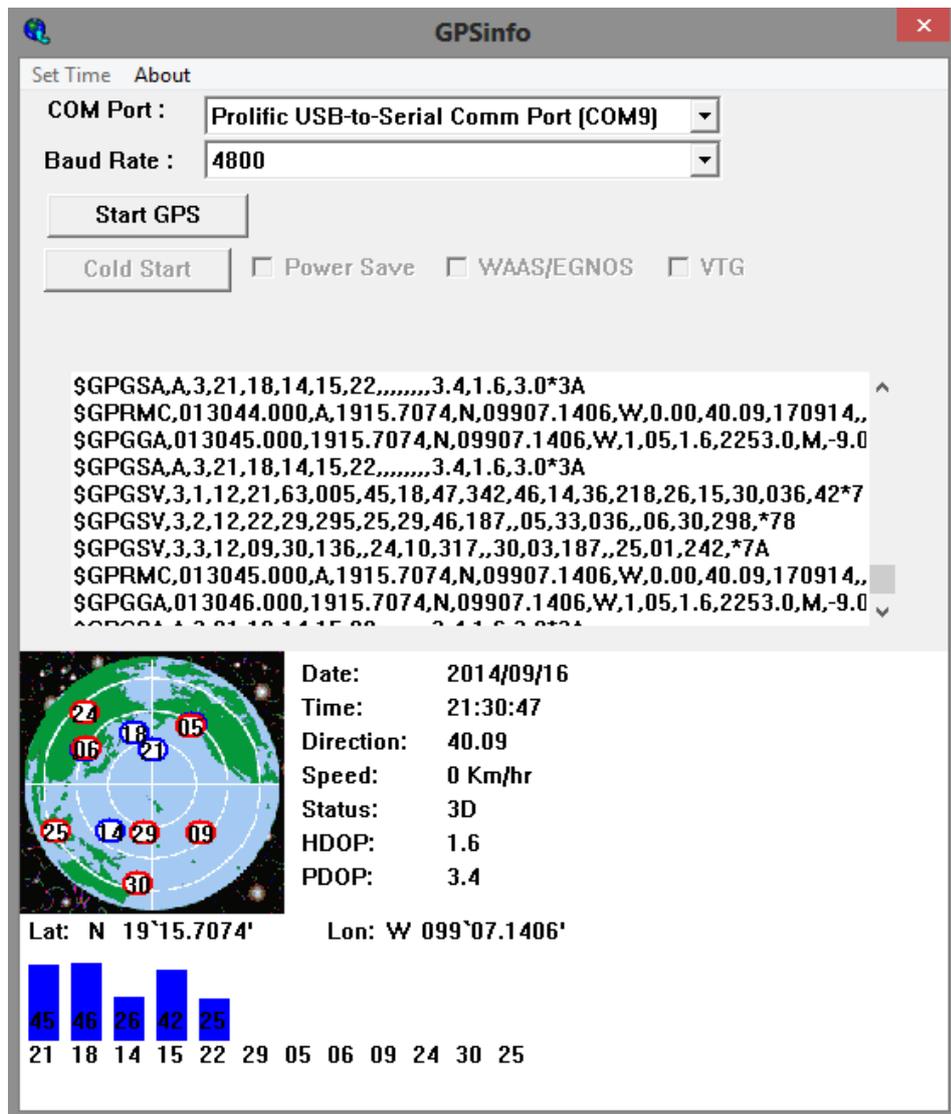
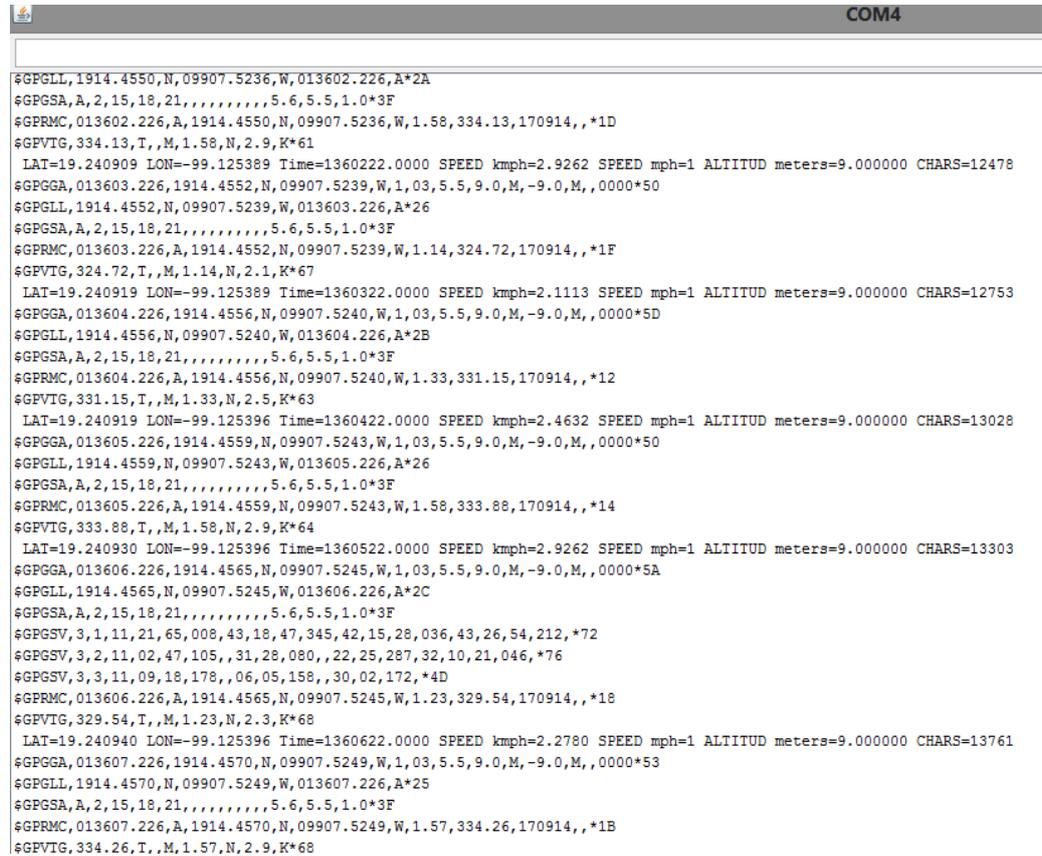


FIGURA 5.20 POSICIÓN GENERADA CON RECEPTOR BU353 INTERPRETADA CON GPS INFO

La cadena GPGA obtenida con el receptor BU353 es la siguiente:

```
$GPGGA,013045.000,1915.7074,N,09907.1406,W,1,05,1.6,2253.0,M,-9.0,M,,0000*61
```

Quitando el comentario para ocultar todas las cadenas recibidas del código GPS_iTead_parameters.ino se tiene un monitor serial como el mostrado en la figura 5.21



```
COM4
$GPGLL,1914.4550,N,09907.5236,W,013602.226,A*2A
$GPGSA,A,2,15,18,21,,,,,,,,,5.6,5.5,1.0*3F
$GPRMC,013602.226,A,1914.4550,N,09907.5236,W,1.58,334.13,170914,,*1D
$GPVTG,334.13,T,,M,1.58,N,2.9,K*61
LAT=19.240909 LON=-99.125389 Time=1360222.0000 SPEED kmph=2.9262 SPEED mph=1 ALTITUD meters=9.000000 CHARS=12478
$GPGGA,013603.226,1914.4552,N,09907.5239,W,1,03,5.5,9.0,M,-9.0,M,,0000*50
$GPGLL,1914.4552,N,09907.5239,W,013603.226,A*26
$GPGSA,A,2,15,18,21,,,,,,,,,5.6,5.5,1.0*3F
$GPRMC,013603.226,A,1914.4552,N,09907.5239,W,1.14,324.72,170914,,*1F
$GPVTG,324.72,T,,M,1.14,N,2.1,K*67
LAT=19.240919 LON=-99.125389 Time=1360322.0000 SPEED kmph=2.1113 SPEED mph=1 ALTITUD meters=9.000000 CHARS=12753
$GPGGA,013604.226,1914.4556,N,09907.5240,W,1,03,5.5,9.0,M,-9.0,M,,0000*5D
$GPGLL,1914.4556,N,09907.5240,W,013604.226,A*2B
$GPGSA,A,2,15,18,21,,,,,,,,,5.6,5.5,1.0*3F
$GPRMC,013604.226,A,1914.4556,N,09907.5240,W,1.33,331.15,170914,,*12
$GPVTG,331.15,T,,M,1.33,N,2.5,K*63
LAT=19.240919 LON=-99.125396 Time=1360422.0000 SPEED kmph=2.4632 SPEED mph=1 ALTITUD meters=9.000000 CHARS=13028
$GPGGA,013605.226,1914.4559,N,09907.5243,W,1,03,5.5,9.0,M,-9.0,M,,0000*50
$GPGLL,1914.4559,N,09907.5243,W,013605.226,A*26
$GPGSA,A,2,15,18,21,,,,,,,,,5.6,5.5,1.0*3F
$GPRMC,013605.226,A,1914.4559,N,09907.5243,W,1.58,333.88,170914,,*14
$GPVTG,333.88,T,,M,1.58,N,2.9,K*64
LAT=19.240930 LON=-99.125396 Time=1360522.0000 SPEED kmph=2.9262 SPEED mph=1 ALTITUD meters=9.000000 CHARS=13303
$GPGGA,013606.226,1914.4565,N,09907.5245,W,1,03,5.5,9.0,M,-9.0,M,,0000*5A
$GPGLL,1914.4565,N,09907.5245,W,013606.226,A*2C
$GPGSA,A,2,15,18,21,,,,,,,,,5.6,5.5,1.0*3F
$GPGSV,3,1,11,21,65,008,43,18,47,345,42,15,28,036,43,26,54,212,*72
$GPGSV,3,2,11,02,47,105,,31,28,080,,22,25,287,32,10,21,046,*76
$GPGSV,3,3,11,09,18,178,,06,05,158,,30,02,172,*4D
$GPRMC,013606.226,A,1914.4565,N,09907.5245,W,1.23,329.54,170914,,*18
$GPVTG,329.54,T,,M,1.23,N,2.3,K*68
LAT=19.240940 LON=-99.125396 Time=1360622.0000 SPEED kmph=2.2780 SPEED mph=1 ALTITUD meters=9.000000 CHARS=13761
$GPGGA,013607.226,1914.4570,N,09907.5249,W,1,03,5.5,9.0,M,-9.0,M,,0000*53
$GPGLL,1914.4570,N,09907.5249,W,013607.226,A*25
$GPGSA,A,2,15,18,21,,,,,,,,,5.6,5.5,1.0*3F
$GPRMC,013607.226,A,1914.4570,N,09907.5249,W,1.57,334.26,170914,,*1B
$GPVTG,334.26,T,,M,1.57,N,2.9,K*68
```

FIGURA 5.21 MONITOR SERIAL CON TODAS LAS CADENAS DE DATOS RECIBIDAS POR ITEAD SHIELD

De la figura 5.21 tenemos que la cadena GPGA recibida por el receptor iTead es la siguiente:

```
$GPGGA,013606.226,1914.4565,N,09907.5245,W,1,03,5.5,9.0,M,-9.0,M,,0000*5A
```

Se puede concluir que la medición generada con el receptor iTeard Studio es correcta pues únicamente hay una variación de pocos grados en latitud y longitud, además de haber generado un mapa en Pixelis con una posición con una variación de aproximadamente 5 metros respecto a la posición real generada con Google maps.

Prototipo de Módulo GPS

Para validar el diseño de la PCB presentada en el capítulo 4 Integración del Sistema Simplificado, se implementó un prototipo (figura 5.23) que emuló las mismas conexiones contempladas en la PCB (figura 5.22).

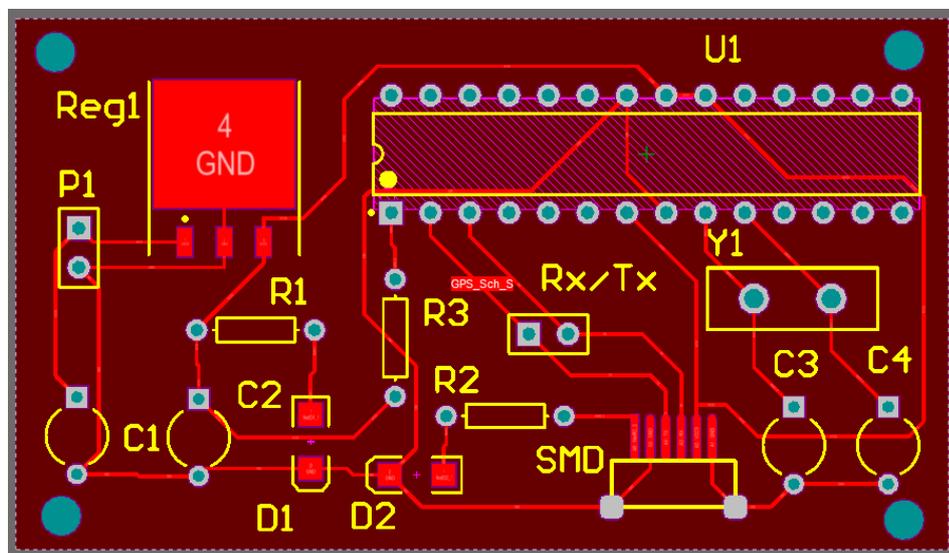


FIGURA 5.22 DISEÑO PROPUESTO DE PCB, MÓDULO RECEPTOR GPS

Las dimensiones físicas así como los esquemáticos de conexiones están disponibles en el capítulo 4.

La única variación que se realizó respecto al sistema propuesto en el capítulo cuatro es que para probar el funcionamiento del prototipo se utilizó el receptor iTeard Studio en lugar del receptor EM-506

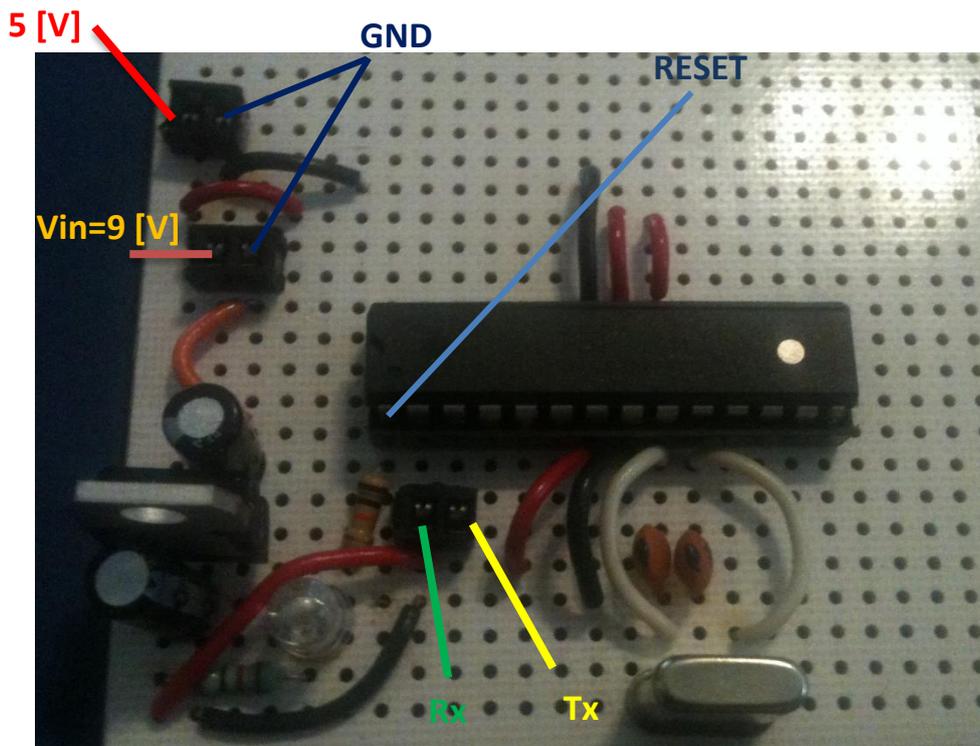


FIGURA 5.23 PROTOTIPO DE MÓDULO GPS

La identificación de puntos de prueba de la PCB se pueden observar en la figura 5.23

Para probar el prototipo se realizó la configuración que se describirá a continuación. Este sistema contiene los elementos mostrados en la tabla 5.1

Componentes
Regulador 5 [V]
Microcontrolador Atmega328
Tarjeta arduino UNO sin microcontrolador
Cristal de 16 [MHz]
Capacitor de 22 [pF]
Capacitor de 10 [uF]
Resistencia de 10 [KΩ]
Resistencia de 220 [KΩ]
Resistencia de 1 [KΩ]
Diodo LED
Receptor iTeed Studio
Antena Externa

TABLA 5.1 COMPONENTES DEL PROTOTIPO DE MÓDULO GPS

Para poder visualizar los datos recibidos mediante el prototipo implementado se requiere una tarjeta arduino uno sin microcontrolador ATmega328 como el mostrado en la figura 5.24 ya que en el prototipo de módulo GPS se ha eliminado el microcontrolador ATmega16U2 que es utilizado para realizar la transferencia de información entre la computadora y nuestra tarjeta de desarrollo convirtiendo de interfaz 232 a USB y viceversa.



FIGURA 5.24 ARDUINO UNO SIN MICROCONTROLADOR ATMEGA328

Para realizar la comunicación entre el prototipo y la computadora se utilizará el microcontrolador ATmega16U2 integrado en la tarjeta Arduino. Es de suma importancia remover el microcontrolador ATmega328 de la tarjeta arduino UNO para dejar que el microprocesador del prototipo sea el que realice todo el trabajo de procesamiento de datos y que este pueda desempeñar su función de manera óptima.

Las conexiones mostradas en la figura 5.25 son requeridas para hacer uso del microcontrolador ATmega16U2 y lograr la comunicación entre el prototipo y la computadora.

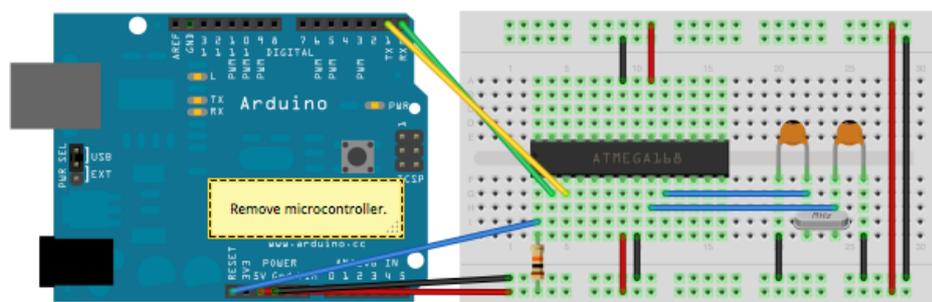


FIGURA 5.25 DIAGRAMA DE CONEXIONES REQUERIDAS PARA COMUNICAR PROTOTIPO GPS CON COMPUTADORA MEDIANTE TARJETA ARDUINO UNO

Es importante señalar que el prototipo puede ser energizado de dos maneras:

- 1) La primera es utilizar una fuente externa como una pila de 9 [V] colocada en los pines Vin y GND la que estará conectada a un regulador de voltaje que nos energizará el microcontrolador con 5 [V] para su correcto funcionamiento.
- 2) La segunda opción es utilizar los 5 [V] que entrega la tarjeta arduino directamente al prototipo. Muy útil cuando no se cuenta con una fuente externa.

Además de las conexiones mostradas en la figura 5.25 se deberá energizar el shield iTeed Studio y conectar los pines de transmisión y recepción de datos TX y Rx respectivamente a los puntos de prueba TX y RX situados en el prototipo.

El sistema con todas las conexiones requeridas es mostrado en la figura 5.26

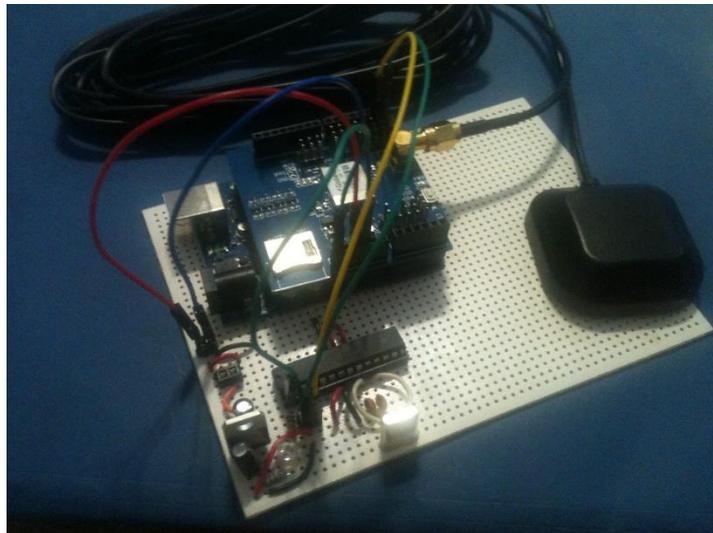


FIGURA 5.26 PROTOTIPO DE MÓDULO GPS CONECTADO A RECEPTOR ITEAD STUDIO CON ANTENA EXTERNA Y ARDUINO UNO

Para cargar el programa en el microcontrolador ATmega328 se siguieron los pasos descritos en el capítulo 4.

El mismo programa utilizado para probar el receptor iTeed Studio, GPS_iTeed_parameters.ino fue cargado en el microcontrolador con el que cuenta el prototipo.

Se optó por tomar este programa ya que es fácil de modificar para obtener los parámetros que el sistema de monitoreo requiera.

Para visualizar los datos recibidos por el prototipo de altitud, latitud, longitud, tiempo y velocidad se deberá conectar la tarjeta arduino a la computadora y abrir el monitor serial configurado a 115200 baudios.

Se realizó una prueba similar a la realizada con el receptor iTeed y se obtuvo el monitor serial de la figura 5.27

```

COM6
LAT=19.324020 LON=-99.166961 Time=16254700.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=185803
LAT=19.324020 LON=-99.166961 Time=16254800.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=186069
LAT=19.324020 LON=-99.166961 Time=16254900.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=186490
LAT=19.324020 LON=-99.166961 Time=16255000.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=186799
LAT=19.324020 LON=-99.166961 Time=16255100.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=187065
LAT=19.324020 LON=-99.166961 Time=16255200.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=187331
LAT=19.324020 LON=-99.166961 Time=16255300.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=187597
LAT=19.324020 LON=-99.166961 Time=16255400.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=188063
LAT=19.324020 LON=-99.166961 Time=16255500.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=188329
LAT=19.324020 LON=-99.166961 Time=16255600.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=188595
LAT=19.324020 LON=-99.166961 Time=16255700.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=188861
LAT=19.324020 LON=-99.166961 Time=16255800.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=189127
LAT=19.324020 LON=-99.166961 Time=16255900.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=189591
LAT=19.324020 LON=-99.166961 Time=16260000.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=189857
LAT=19.324020 LON=-99.166961 Time=16260100.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=190123
LAT=19.324020 LON=-99.166961 Time=16260200.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=190389
LAT=19.324020 LON=-99.166961 Time=16260300.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=190655
LAT=19.324020 LON=-99.166961 Time=16260400.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=191119
LAT=19.324020 LON=-99.166961 Time=16260500.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=191385
LAT=19.324020 LON=-99.166961 Time=16260600.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=191651
LAT=19.324020 LON=-99.166961 Time=16260700.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=191917
LAT=19.324020 LON=-99.166961 Time=16260800.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=192183
LAT=19.324020 LON=-99.166961 Time=16260900.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=192647
LAT=19.324020 LON=-99.166961 Time=16261000.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=192913
LAT=19.324020 LON=-99.166961 Time=16261100.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=193179
LAT=19.324020 LON=-99.166961 Time=16261200.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=193445
LAT=19.324020 LON=-99.166961 Time=16261300.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=193711
LAT=19.324020 LON=-99.166961 Time=16261400.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=194175
LAT=19.324020 LON=-99.166961 Time=16261500.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=194441
LAT=19.324020 LON=-99.166961 Time=16261600.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=194707
LAT=19.324020 LON=-99.166961 Time=16261700.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=194973
LAT=19.324020 LON=-99.166961 Time=16261800.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=195239
LAT=19.324020 LON=-99.166961 Time=16261900.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=195705
LAT=19.324020 LON=-99.166961 Time=16262000.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=195971
LAT=19.324020 LON=-99.166961 Time=16262100.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=196237
LAT=19.324020 LON=-99.166961 Time=16262200.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=196503
LAT=19.324020 LON=-99.166961 Time=16262300.0000 SPEED kmph=0.0000 SPEED mph=0 ALTIITUD meters=73.599998 CHARS=196769
 Desplazamiento automático

```

FIGURA 5.27 DATOS DE POSICIÓN OBTENIDOS MEDIANTE PROTOTIPO DEL MÓDULO RECEPTOR GPS

Y para validar los datos obtenidos se ingresaron la latitud y longitud en Pixelis Maps generando el mapa que se muestra en la figura 5.28

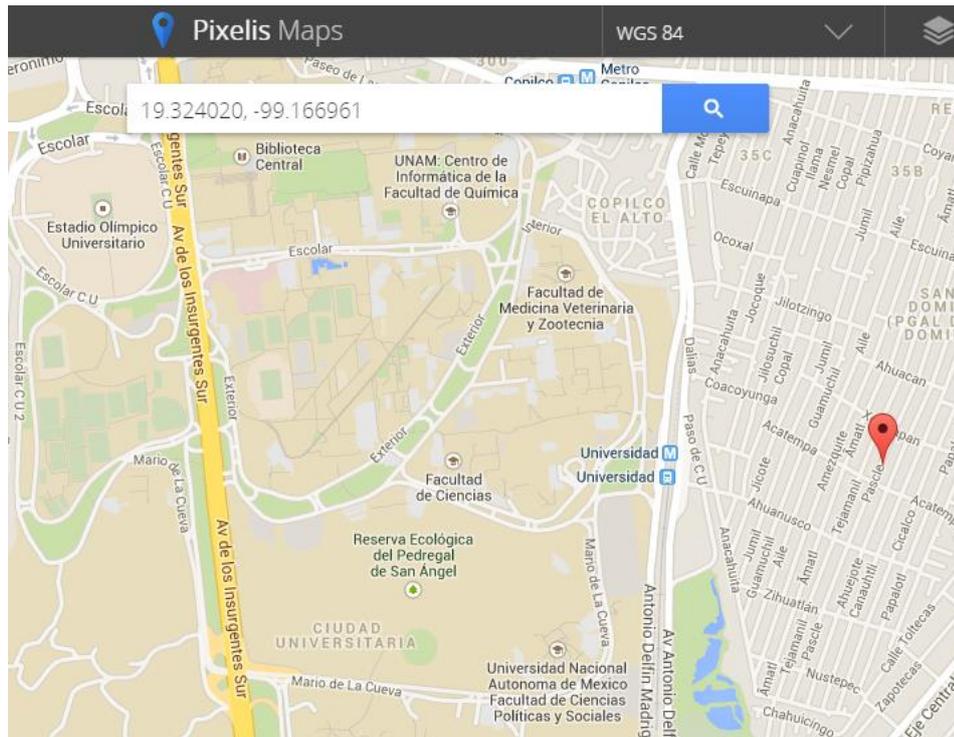


FIGURA 5.28 MAPA GENERADO POR PÍXELES MAPS CON LA POSICIÓN OBTENIDA CON EL PROTOTIPO DEL MÓDULO GPS

Para validar esta posición se comparó con la obtenida en Google maps (figura 5.29) teniendo una diferencia de pocos metros respecto una posición y otra.

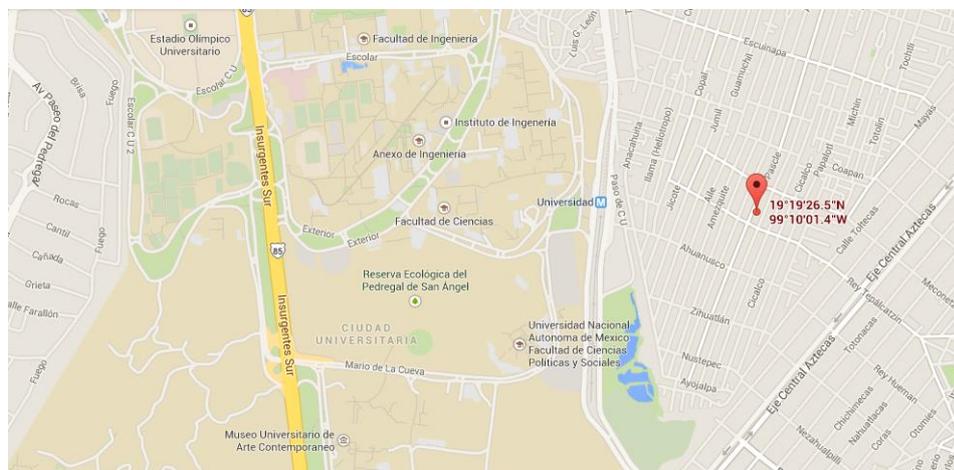


FIGURA 5.29 POSICIÓN Y MAPA OBTENIDOS CON GOOGLE MAPS

Por último se realizó una prueba de recepción en interiores arrojando el resultado que se muestra en el monitor serial de la figura 5.30

```
CHARS=41174
LAT=19.265729 LON=-99.127876 Time=45693.0000 SPEED kmph=3.0188 SPEED mph=1 ALTIITUD meters=9.000000 CHARS=41449
CHARS=41683
CHARS=41917
CHARS=42319
CHARS=42579
LAT=19.265619 LON=-99.127822 Time=50193.0000 SPEED kmph=1.2594 SPEED mph=0 ALTIITUD meters=9.000000 CHARS=42854
CHARS=43088
CHARS=43322
CHARS=43724
CHARS=43984
LAT=19.265640 LON=-99.127952 Time=50693.0000 SPEED kmph=8.8155 SPEED mph=5 ALTIITUD meters=9.000000 CHARS=44257
CHARS=44491
CHARS=44725
CHARS=45135
CHARS=45387
LAT=19.265619 LON=-99.127922 Time=51193.0000 SPEED kmph=12.0936 SPEED mph=7 ALTIITUD meters=9.000000 CHARS=45661
CHARS=45895
CHARS=46129
CHARS=46547
CHARS=46789
CHARS=47023
CHARS=47257
CHARS=47491
CHARS=47917
CHARS=48151
LAT=19.265590 LON=-99.128028 Time=52193.0000 SPEED kmph=1.1112 SPEED mph=0 ALTIITUD meters=9.000000 CHARS=48426
CHARS=48660
```

FIGURA 5.30 MONITOR SERIAL CON DATOS RECIBIDOS SITUADOS EN INTERIOR CON PROTOTIPO DE MÓDULO GPS

El receptor fue capaz de determinar la posición pese a no tener línea de vista, lo que provoca que se tarde mucho más tiempo en determinar la posición (2 minutos 57 segundos) comparado a cuando se tiene línea de vista y buen tiempo (20 segundos).

Además se aprecia que no todas las cadenas que se recibieron estuvieron completas puesto que no con todas fue posible determinar una posición.

Estas consecuencias no son producidas únicamente por la falta de línea de vista sino que también al estar en un interior el nivel de potencia recibido se ve afectado lo que dificulta la tarea de recepción.

Conclusiones

El módulo receptor GPS desarrollado en este proyecto permitirá otorgar la funcionalidad de geo localización a un sistema de monitoreo de parámetros físicos configurado con una cantidad de componentes mínima, de dimensiones reducidas para facilitar el proceso de integración con otros subsistemas y con una precisión de pocos metros dependiendo del número de satélites que se tenga en línea de vista así como de las condiciones climáticas en las que se encuentre.

Gracias al desarrollo realizado en este proyecto y al microcontrolador con el funciona el módulo GPS, el sistema posee un alto nivel de configuración que será de gran utilidad para cubrir las diferentes necesidades que requiera el sistema al que se incorpore. Esto resulta en una gran ventaja ya que el módulo GPS propuesto podrá ser adaptado a sistemas con diversas aplicaciones.

Además de otorgar la posición, este sistema es capaz de otorgar altitud, velocidad en kilómetros por hora o millas por hora, curso en grados, número de satélites en línea de vista, tiempo local y tiempo de estado fix.

Una de las aplicaciones que podría tener el módulo GPS sería incorporarlo a un sistema de telemetría para monitorear la presión y gasto volumétrico de tanques de gas. Lo anterior permitiría implementar un sistema automatizado con el que se lograría detectar posibles anomalías en la presión de los tanques para brindar mantenimiento antes de presentarse algún incidente, además de monitorear el gasto de gas realizado.

Todos los datos recabados se podrían concentrar y desplegar en una aplicación donde ya sea la compañía de gas o los mismos usuarios pudieran obtener información sobre el estado del tanque así como del gasto que se ha realizado dentro de cierto intervalo de tiempo.

Incorporar el módulo GPS al sistema de telemetría sería de gran utilidad ya que si se conoce la posición de donde se obtienen las lecturas ya no nos veríamos en la necesidad de crear, implementar y consultar un base de datos con la que se pudiera relacionar las diferentes mediciones obtenidas con una ubicación geográfica reduciendo el uso de recursos humanos. En caso de ser requerido un ajuste manual será posible direccionar de una manera rápida y sencilla al personal pertinente ya que se podrá discriminar entre los múltiples elementos que conformarían el sistema de telemetría.

El módulo diseñado también podría incorporarse a un sistema de telemetría para monitorear el estado de un satélite siempre y cuando la órbita de este sea inferior a la órbita de la constelación GPS ya que conocer la posición del satélite facilitaría la tarea de intercambio de información o comandos con la estación de control.

Otro proyecto en donde este módulo GPS podría ser integrado, es en el sistema de adquisición de datos desarrollado por alumnos de la UNAM miembros de la escudería Unam Motorsports quienes participan en la competencia Formula SAE (Sociedad de Ingenieros Automotrices).

De manera superficial, en este proyecto se diseña y construye un carro desde su carrocería hasta el sistema de suspensión. En la última temporada decidió implementarse un sistema de adquisición de datos para recabar información sobre los siguientes parámetros: Velocidad, aceleraciones y giros en los tres ejes, temperatura de la tarjeta de adquisición, temperatura del motor, rpm del motor, compresión de los resortes, ángulo del volante, ángulo del pedal de freno, posición del acelerador y presiones de las dos líneas de freno.

Lo anterior permitió no solo monitorear el desempeño del carro en la temporada 2013-2014 sino además permitirá mejorar el desempeño general del vehículo así como del piloto en futuras temporadas.

Integrar el módulo GPS al sistema de adquisición de datos podría resultar en varias ventajas ya que se podrían asociar los datos recabados a una posición de la pista con lo que se podría determinar de una mejor manera el desempeño del carro al generar gráficas de datos con respecto a la distancia permitiendo separar los datos de las diferentes vueltas realizadas.

El módulo GPS también permitiría realizar un estimado de la velocidad, indicar el curso y realizar una correlación del trazado de la pista con los tiempos de vuelta, buen indicador para saber si el piloto está tomando las curvas de manera correcta.

Como podemos observar la funcionalidad del módulo GPS irá muy de la mano con las aplicaciones de los sistemas a los que se desee incorporarlo.

Por otro lado, como se pudo notar en las pruebas realizadas a lo largo de esta tesis, el sistema propuesto a pesar de sufrir cierta vulnerabilidad con las condiciones climáticas o con la visibilidad con la que cuente el receptor, será capaz de determinar la posición siempre y cuando se cumpla que la señal recibida tenga un nivel de potencia igual o mayor al mínimo aceptado por el receptor. Lo anterior nos muestra que el desempeño del sistema estará fuertemente relacionado con las especificaciones técnicas del receptor por lo

que se espera que con el paso del tiempo, este sistema se pueda equipar con un receptor con mejores características a un precio más económico.

Otro aspecto que se podría mejorar es el tamaño de algunos de los componentes utilizados. Se podrían sustituir los empaquetados tradicionales de las resistencias y capacitores por un empaquetado de montaje superficial permitiendo reducir de manera considerable el tamaño de la tarjeta diseñada en Altium Designer 14.

Pese a las diferentes mejoras que pueden realizarse sobre el sistema presentado, con este proyecto se aporta un conjunto de experiencias suficientes que podrán utilizarse como un modelo base con el que se podrá desarrollar tecnología enfocada hacia el diseño de sistemas similares.

Gracias a sus elementos sencillos y de fácil implementación, se propuso un sistema de bajo coste que estará al alcance de la mayoría de instituciones educativas y de investigación, así como de particulares que necesiten disponer del sistema.

Es importante remarcar que gracias al código reconfigurable desarrollado, el sistema podrá ajustarse fácilmente a las necesidades de diferentes proyectos similares por lo que este podrá ser reutilizado tal y como se pudo observar en los diferentes sistemas utilizados en este proyecto.

Anexos

GPS_SIMPLE_TEST_EM408.INO

```
/* Este ejemplo ayuda a comprobar el correcto funcionamiento del receptor
   GPS EM-408 el cual requiere el uso de SoftwareSerial, y se asume que
   se tiene el dispositivo GPS conectado a pins 0(rx) and 1(tx) a 4800-baud.
*/

#include <SoftwareSerial.h> //Se agrega librería SoftwareSerial
#include <TinyGPS.h>       //Se agrega librería TinyGPS
TinyGPS gps;
SoftwareSerial ss(0, 1); //Declaración de puertos seriales en pines digitales 0 y 1

void setup()
{
  Serial.begin(115200); //Velocidad del monitor serial
  ss.begin(4800); //Velocidad del receptor GPS

  Serial.print("Prueba para receptor GPS EM-408 y libreria TinyGPS version: ");
  Serial.println(TinyGPS::library_version()); //Muestra información del programa y
  versión de librería TinyGPS
  Serial.println("Modificado por Roberto Ortega");
  Serial.println();
}

void loop()
{
```

```

bool newData = false;    //Se declara la variable newData con valor booleano
falso

unsigned long chars;    //Se declara la variable de tipo long chars
unsigned short sentences, failed; //Se declaran las variables de tipo short
sentences y failed

// Se buscan datos cada segundo por los valores clave GPS mediante ciclo for
for (unsigned long start = millis(); millis() - start < 1000;)
{
    while (ss.available())
    {
        char c = ss.read();
        // Serial.write(c); // Quitar comentario para visualizar los datos GPS recibidos
        if (gps.encode(c)) // Busca si entra otra sentencia válida
            newData = true;
    }
}

if (newData) // En caso de encontrar datos válidos
{
    float flat, flon;
    unsigned long age;
    gps.f_get_position(&flat, &flon, &age); //Obtiene la posición
    Serial.print("LAT="); //Imprime serialmente la Latitud
    Serial.print(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat, 6);
    Serial.print(" LON="); //Imprime serialmente la Longitud
    Serial.print(flou == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flon, 6);
    Serial.print(" SAT="); //Imprime serialmente el número de satélites
}

```

```

Serial.print(gps.satellites() == TinyGPS::GPS_INVALID_SATELLITES ? 0 :
gps.satellites());
Serial.print(" PREC="); //Imprime serialmente la precisión
Serial.print(gps.hdop() == TinyGPS::GPS_INVALID_HDOP ? 0 : gps.hdop());
}

```

```

gps.stats(&chars, &sentences, &failed); //Obtiene las estadísticas de los datos
recibidos

```

```

Serial.print(" CHARS="); //Imprime el número de caracteres recibidos
Serial.print(chars);
Serial.print(" SENTENCES="); //Imprime el número de sentencias recibidas
Serial.print(sentences);
Serial.print(" CSUM ERR="); //Imprime el número de sentencias en error
recibidas
Serial.println(failed);
}

```

GPS_LOGGER_MEGA.INO

```

// Roberto Ortega GPS Logger
#include <SD.h> //Se agrega la librería SD
#include <stdlib.h> //Se agrega la librería stdlib
#include <TinyGPS.h> //Se agrega la librería TinyGPS

```

```

/* Codigo GPS Logger para arduino MEGA
Se necesita Wireless SD Shield y un
dispositivo GPS a 4800-baud
conectado al Serial3 pins 15(rx) and 14(tx).
*/

```

```

TinyGPS gps;

```

```

static char dtostrfbuffer[20];
int CS = 4; //Pin designado en el shield para CS (SD card).

int LED = 8; //Indicador

//Definiendo las cadenas para almacenarlas en la SD
String SD_date_time= "invalid"; //Cadena para almacenar tiempo
String SD_lat = "invalid"; //Cadena para almacenar latitud
String SD_lon = "invalid"; //Cadena para almacenar longitud
String dataString = "";

//Se declaran las funciones utilizadas en el programa
static void gpstdump(TinyGPS &gps);
static bool feedgps();
static void print_float(float val, float invalid, int len, int prec, int SD_val);
static void print_int(unsigned long val, unsigned long invalid, int len);
static void print_date(TinyGPS &gps);
static void print_str(const char *str, int len);

void setup()
{

    pinMode(53, OUTPUT); //Para un correcto funcionamiento de la SD 10
    arduino uno 53 mega
    pinMode(LED,OUTPUT); //Indicador LED

    //Interfaces seriales
    Serial.begin(115200); //Velocidad monitor serial
    Serial3.begin(4800); //Velocidad del receptor GPS

    //Conectando a la SD card
    if(!SD.begin(CS))
    {
        Serial.println("SD card no encontrada"); //Indicador de tarjeta SD no
        encontrada
        return; //Rompe el ciclo
    }
    //Para visualizar de manera más fácil en el monitor serial
    Serial.print(" TinyGPS library v. "); Serial.println(TinyGPS::library_version());
    //Muestra versión de librería TinyGPS

```

```

Serial.println("Roberto Ortega GPS DATA LOGGER");
Serial.println();
Serial.print("Sizeof(gpsobject) = "); Serial.println(sizeof(TinyGPS));
Serial.println();
Serial.println("Sats HDOP Latitude Longitude Fix Date    Time    Date Alt
Course Speed Card  Distance Course Card  Chars Sentences Checksum");
//Encabezados de datos
Serial.println("    (deg) (deg) Age          Age (m)  --- from GPS --
-- ---- to London ---- RX  RX    Fail");
Serial.println("-----");
-----");
}

void loop()
{
  bool newdata = false;
  unsigned long start = millis();

  // Ciclo para recibir datos GPS cada segundo
  while (millis() - start < 1000)
  {
    if (feedgps())
      newdata = true;
  }

  gpsdump(gps); //Inicia la función gpsdump

  //Grabando informacion en la SD
  dataString = SD_date_time + "," + SD_lat + "," + SD_lon;
  if(SD_date_time != "invalid")
    digitalWrite(LED,HIGH); //En caso de no encontrar tarjeta SD el indicador
LED se encenderá
  else
    digitalWrite(LED,LOW); //Si no hay problema con la tarjeta SD el indicador
permanece apagado

  //Abriendo archivo CSV en la tarjeta SD
  File dataFile = SD.open("LOG.csv" , FILE_WRITE); //Si se desea un archivo .txt
se deberá cambiar la extensión en esta línea
  if (dataFile)
  {

```

```

        dataFile.println(dataString); //Si logra acceder a la tarjeta SD graba la
información
        Serial.println(dataString); //Se imprime de manera serial la información
        dataFile.close(); //Cierra el archivo LOG.csv
    }
    else //Despliega serialmente si no es posible acceder a la tarjeta SD
    {
        Serial.println("\n No se pudo acceder al archivo CSV");
    }
}

static void gpsdump(TinyGPS &gps) //Función gpsdump
{
    //Declaración de variables dentro la función gpsdump
    float flat, flon;
    unsigned long age, date, time, chars = 0;
    unsigned short sentences = 0, failed = 0;
    static const float LONDON_LAT = 51.508131, LONDON_LON = -0.128002;

    print_int(gps.satellites(), TinyGPS::GPS_INVALID_SATELLITES, 5); //Imprime el
número de satélites
    print_int(gps.hdop(), TinyGPS::GPS_INVALID_HDOP, 5); //Imprime hdop
    gps.f_get_position(&flat, &flon, &age); //Función para obtener posición
    print_float(flat, TinyGPS::GPS_INVALID_F_ANGLE, 9, 5, 1); //Imprime Latitud
    print_float(flon, TinyGPS::GPS_INVALID_F_ANGLE, 10, 5, 2); //Imprime
Longitud
    print_int(age, TinyGPS::GPS_INVALID_AGE, 5); //Imprime el tiempo de fix

    print_date(gps); //Date and time

    print_float(gps.f_altitude(), TinyGPS::GPS_INVALID_F_ALTITUDE, 8, 2, 0);
//Imprime altitud
    print_float(gps.f_course(), TinyGPS::GPS_INVALID_F_ANGLE, 7, 2, 0);
//Imprime curso
    print_float(gps.f_speed_kmph(), TinyGPS::GPS_INVALID_F_SPEED, 6, 2, 0);
//Imprime velocidad en kmph
    print_str(gps.f_course() == TinyGPS::GPS_INVALID_F_ANGLE ? "*** " :
TinyGPS::cardinal(gps.f_course()), 6);

```

```

    print_int(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0UL : (unsigned
long)TinyGPS::distance_between(flat, flon, LONDON_LAT, LONDON_LON) /
1000, 0xFFFFFFFF, 9); //Imprime distancia a Londres
    print_float(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 :
TinyGPS::course_to(flat, flon, 51.508131, -0.128002),
TinyGPS::GPS_INVALID_F_ANGLE, 7, 2, 0); //Imprime curso a Londres
    print_str(flat == TinyGPS::GPS_INVALID_F_ANGLE ? "*** " :
TinyGPS::cardinal(TinyGPS::course_to(flat, flon, LONDON_LAT, LONDON_LON)),
6);

```

```

    gps.stats(&chars, &sentences, &failed); //Obtiene estadísticas de los datos
recibidos
    print_int(chars, 0xFFFFFFFF, 6); //Imprime el número de caracteres
recibidos
    print_int(sentences, 0xFFFFFFFF, 10); //Imprime el número de sentencias
recibidas
    print_int(failed, 0xFFFFFFFF, 9); //Imprime el número de sentencias con
error recibidas
    Serial.println();
}

```

```

static void print_int(unsigned long val, unsigned long invalid, int len) //Función
print_int utilizada para dar formato de impresión serial a datos de tipo int
{
    char sz[32];
    if (val == invalid)
        strcpy(sz, "*****");
    else
        sprintf(sz, "%ld", val);
    sz[len] = 0;
    for (int i=strlen(sz); i<len; ++i)
        sz[i] = ' ';
    if (len > 0)
        sz[len-1] = ' ';
    Serial.print(sz);
    feedgps();
}

```

```

static void print_float(float val, float invalid, int len, int prec, int SD_val)
//Función print_float utilizada para dar formato de impresión serial a datos de
tipo float

```

```

{
char sz[32];
if (val == invalid) //Secuencia para indicar si se tiene un valor inválido
{
strcpy(sz, "*****");
sz[len] = 0;
if (len > 0)
sz[len-1] = ' ';
for (int i=7; i<len; ++i)
sz[i] = ' ';
Serial.print(sz);
//Se identifica si se trata de long o latit y se guarda en las variables para la SD
if(SD_val == 1) SD_lat = sz;
else if(SD_val == 2) SD_lon = sz;

}
else
{
Serial.print(val, prec); //Imprime de manera serial

if (SD_val == 1) SD_lat = dtostrf(val,10,5,dtostrfbuffer);
else if (SD_val == 2) SD_lon = dtostrf(val,10,5,dtostrfbuffer);
//Ciclo para determinar precisión
int vi = abs((int)val);
int flen = prec + (val < 0.0 ? 2 : 1);
flen += vi >= 1000 ? 4 : vi >= 100 ? 3 : vi >= 10 ? 2 : 1;
for (int i=flen; i<len; ++i)
Serial.print(" ");

}
feedgps(); //Toma siguiente cadena de datos
}

static void print_date(TinyGPS &gps) //Función print_date utilizada para
obtener la fecha, guardarla en la memoria SD y mostrarla serialmente
{
int year;
byte month, day, hour, minute, second, hundredths;
unsigned long age;
gps.crack_datetime(&year, &month, &day, &hour, &minute, &second,
&hundredths, &age);

```

```

if (age == TinyGPS::GPS_INVALID_AGE) //Ciclo para detectar si se tiene una
fecha invalida
{
    Serial.print("*****  ***** ");
    SD_date_time = "invalid";
}
else //Ciclo para mostrar la fecha de manera serial y guardar este valor en la
memoria SD
{
    char sz[32];
    sprintf(sz, "%02d/%02d/%02d %02d:%02d:%02d ",
        month, day, year, hour, minute, second);
    Serial.print(sz);
    //Se guarda el date_time de la SD
    SD_date_time = sz;
}
print_int(age, TinyGPS::GPS_INVALID_AGE, 5);
feedgps();
}

static void print_str(const char *str, int len) //Función print_str que da formato
para imprimir de manera serial las cadenas generadas
{
    int slen = strlen(str);
    for (int i=0; i<len; ++i)
        Serial.print(i<slen ? str[i] : ' ');
    feedgps();
}

static bool feedgps() //Funcion feedgps, lee los datos recibidos por el receptor
en caso de contar con los mismos
{
    while (Serial3.available()) //Si hay datos en el puerto serial 3 que es donde se
encuentra conectado el receptor
    {
        if (gps.encode(Serial3.read())) // Lee los datos del puerto serial 3
            return true;
    }
    return false;
}

```

GPS_ITEAD_PARAMETERS.INO

```
//Roberto Ortega GPS reciever

/* Este ejemplo ayuda a comprobar el correcto funcionamiento de un receptor
GPS el cual requiere el uso de SoftwareSerial y librería Tiny GPS, se asume
que
se tiene el dispositivo GPS conectado a pins 0(rx) and 1(tx) a 9600-baud.
*/
#include <SoftwareSerial.h> //Se inician librerías softwareSerial y TinyGPS
#include <TinyGPS.h>

TinyGPS gps; //Se crea el objeto gps
SoftwareSerial ss(0, 1); //Declaración de puerto serial en pines digitales 0 y 1

void setup()
{
  Serial.begin(115200); //Velocidad del monitor serial
  ss.begin(9600);      //Velocidad del receptor GPS
  //Encabezados seriales para mejor visualización
  Serial.print("Prueba para receptor GPS iTead y libreria TinyGPS version: ");
  Serial.println(TinyGPS::library_version()); //Muestra versión de librería
  TinyGPS
  Serial.println("Modificado por Roberto Ortega");
  Serial.println();
}

void loop()
{
  bool newData = false;
  unsigned long chars;
  unsigned short sentences, failed;

  // Se buscan datos cada segundo por los valores clave GPS
  for (unsigned long start = millis(); millis() - start < 1000;)
  {
    while (ss.available()) //Si hay datos disponibles en el puerto serial
    {
      char c = ss.read(); //Son leidos y guardados en c
```

```

    //Serial.write(c); // Quitar comentario para visualizar los datos GPS
recibidos
    if (gps.encode(c)) // Busca si entra otra sentencia valida
        newData = true;
    }
}

if (newData) // En caso de encontrar datos válidos
{
    float flat, flon;
    unsigned long age, date, time, speed, course;
    gps.f_get_position(&flat, &flon, &age); //Se obtiene la posición
    gps.get_datetime(&date, &time, &age); //Se obtiene tiempo y fecha
    speed = gps.speed(); //Obtiene velocidad
    int fspeed = gps.f_speed_mph();
    float falt = gps.f_altitude(); // +/- Altitud en metros
    float fc = gps.f_course(); // Curso en grados
    float fkmph = gps.f_speed_kmph(); // Velocidad en km/hr
    /*
Despliegue de datos de manera serial
Para configurar y obtener los datos que se deseen basta con
quitar el parámetro deseado de la sección de comentarios
Igualmente si se desea quitar algún parámetro que se este
mostrando actualmente bastará con comentarlo
*/
    Serial.print(" LAT="); //Latitud
    Serial.print(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat, 6);
    Serial.print(" LON="); //Longitud
    Serial.print(flon == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flon, 6);
    Serial.print(" Time="); //Tiempo
    Serial.print(time == TinyGPS::GPS_INVALID_DATE ? 0.0 : time, 4);
    Serial.print(" SPEED kmph="); //Velocidad en kmph
    Serial.print(fkmph == TinyGPS::GPS_INVALID_F_SPEED ? 0.0 : fkmph, 4);
    Serial.print(" SPEED mph="); //Velocidad en mph
    Serial.print(fspeed);
    Serial.print(" ALTITUD meters="); //Altitud en metros
    Serial.print(falt == TinyGPS::GPS_INVALID_F_ALTITUDE ? 0.0 : falt, 6);

    /*
    Serial.print(" SAT="); //Numero de satélites

```

```

    Serial.print(gps.satellites() == TinyGPS::GPS_INVALID_SATELLITES ? 0 :
gps.satellites());
    Serial.print(" PREC="); //Precisión
    Serial.print(gps.hdop() == TinyGPS::GPS_INVALID_HDOP ? 0 : gps.hdop());
    Serial.print(" AGE="); //Tiempo de tener fix
    Serial.print(age == TinyGPS::GPS_INVALID_AGE ? 0.0 : age, 6);
    Serial.print(" DATE="); //Fecha
    Serial.print(date == TinyGPS::GPS_INVALID_DATE ? 0.0 : date, 6);
    Serial.print(" CURSO [DEGREES]="); //Curso
    Serial.print(fc);
    Serial.print(" SPEED knot="); //Velocidad en knot
    Serial.print(speed);
    Serial.print(" SPEED kmph2=");
    Serial.print(fspeed == TinyGPS::GPS_INVALID_F_SPEED ? 0.0 : fspeed, 6);
    Serial.print(" ALTITUD2=");
    Serial.print(gps.altitude());
    */
}

gps.stats(&chars, &sentences, &failed); //Obtiene estadísticas de los datos
recibidos
Serial.print(" CHARS="); //Imprime de manera serial el número de
caracteres recibidos
Serial.println(chars);
/*Serial.print(" SENTENCES="); //Se comenta el número de sentencias y
errores ya que basta con observar el número de caracteres recibidos para
corroborar el funcionamiento del receptor
Serial.print(sentences);
Serial.print(" CSUM ERR=");
Serial.println(failed);*/
}

```

RECEPTOR BU353



El BU353 es un receptor GPS USB que incorpora una alta sensibilidad, chipset de bajo consumo de poder con un empaquetamiento ultra compacto. El receptor BU353 incorpora un chipset SiRF Star IV el cual provee un desempeño superior en cañones urbanos o en entornos con alto follaje. Con la tecnología SiRF CGEE por sus siglas en inglés (Client Generated Extended Ephemeris), el receptor es capaz de predecir las posiciones de los satélites hasta por tres días permitiéndole realizar un tiempo de inicio de menos de 15 segundos bajo cualquier circunstancia climática sin asistencia de red.

El receptor BU353 dispone del modo MicroPower con el que puede permanecer en estado de arranque rápido continuamente haciendo uso de muy poca cantidad de energía.

Principales características técnicas:

- SiRF star IV
- Restreo con “Todos en vista” -48 Canales
- Compatible con NMEA 0183
- Soporta WAAS/EGNOS
- Diametro de 2.08” x 0.75”
- Antena GPS de parche incluida
- Superficie de montaje magnética incluida
- Interfaz USB 2.0

General Specifications

GPS Chipset.....	SiRF Star IV
Frequency.....	N/A
C/A Code.....	N/A
Channels.....	48 all-in-view tracking
Sensitivity.....	-163 dBm
Accuracy.....	N/A

Aquisition Rate

Hot start.....	1 sec., average
Warm start.....	35 sec., average
Cold start.....	35 sec., average
Reacquisition.....	0.1 sec. average
Update rate.....	1Hz

Protocol

GPS Protocol.....	Default: NMEA 0183 (Secondary: SiRF binary)
GPS Output Data.....	SiRF binary >> position, velocity, altitude, status and control NMEA 0183 V3.0 protocol (GGA, GSA, GSV, RMC, VTG, GLL, ZDA)
GPS transfer rate.....	Software command setting (Default : 4800,n,8,1 for NMEA)

Dynamic Condition

Acceleration Limit.....	Less than 4g
Altitude Limit.....	Max 60,000 Feet (18,000 meters)
Velocity Limit.....	Max 515 Meters / sec (1000 Knots / 151 MPH)
Jerk Limit.....	20 m/sec**3

Environmental Specifications

Operating.....-40°~ 185°F (-40°~ 85°C)
Storage.....-40°~ 185°F (-40°~ 85°C)
Humidity.....Up to 95% non-condensing

Electrical Characteristics

Voltage.....4.5V - 5.5V
Current.....55mA Maximum

Physical Characteristics

Dimension.....2.08" diameter x 0.75" (53mm dia. x 19.2mm)
USB Cable Length.....59" (1,500 mm)
Weight.....2.2Ounces (62.37g)

Carton Specifications

Quantity Per Box.....50
Carton Weight.....26lbs
Carton Dimensions.....29" x 16" x 12" (736.6mm x 406.4mm x 304.8mm)

Acrónimos

A/D (Analogic to Digital Converter) Convertidor Analógico Digital
AFSCN (Air Force Satellite Control Network) Red de Control Satelital de la Fuerza Aerea
Bps Bits por Segundo
CMOS (Complementary metal-oxide Semiconductor) Semiconductor oxido-metal complementario
EGNOS (European Geostationary Navigation Overlay Service) Servicio Superpuesto de Navegación Geoestacionario Europeo
GEO (Geostationary Earth Orbit) Órbita Terrestre Geoestacionaria
GNSS (Global Navigation Satellite System) Sistema global de navegación por satélite
GPS (Global Positioning System) Sistema de posicionamiento global
ICSP (In Circuit Serial Programming) Programación Serial en Circuito
IDE (Integrated Development Environment) Entorno de Desarrollo Integrado
LED (Light Emitting Diode) Diodo Emisor de Luz
LEO (Low Earth Orbit) Órbita Terrestre Baja
MOSI (Master Out Slave In) Salida de Maestro Entrada de Esclavo
MISO (Master In Slave Out) Entrada de Maestro Salida de Esclavo
MSAS (Multifunctional Satellite Augmentation System) Sistema multifuncional de Aumentación por Satélite
NAVSAT (Navy Navigation Satellite System) Sistema de Marina de Navegación por Satélite
NMEA (National Marine Electronics Association) Asociación Nacional de Electronica Marina
PCB (Printed Circuit Board) Tarjeta de circuito impreso
RF Radio Frecuencia
SBAS (Satellite Based Augmentation System) Sistema de aumentación Basado en Satélite
SMA (Surface Mount Assembly Connector) Conector de Montaje Superficial
SMD (Surface Mount Device) Dispositivo de Montaje Superficial
USB (Universal Serial Bus) Puerto Serial Universal
WAAS (Wide Area Augmentation System) Sistema de Aumentación de Área Amplia
2SOPS (Second Space Operations Squadron) Segundo Escuadrón de Operaciones Espaciales

Bibliografía

Altium Designer. "Online Documentation Altium 14." *Getting Started with Altium Designer*. Web. 10 Feb. 2014. <<http://www.altium.com/>>.

China Satellite Navigation Office (CSNO). "Navigation Satellite System." *BeiDou*. Web. 23 Jan. 2014.

Doc. 9849 OACI. "Manual sobre el Sistema Mundial de Navegación por Satélites (GNSS)". 2005

Federal Space Agency, GLONASS-IAC. "GLONASS Constellation Status." *Информационно-аналитический центр контроля ГЛОНАСС и GPS*. Web. 17 Feb. 2014. <<http://glonass-iac.ru/>>.

Félix Pérez Martínez, Universidad Politécnica de Madrid. *Sistemas de Radionavegación. "Sistemas de Navegación por Satélite"*. Apuntes. 2014.

Global Positioning System. Washington, D.C.: U.S. Air Force, 2007. Print.

Google Inc. "Google Earth." Web. 21 May 2014. <<http://earth.google.com/>>.

Google Inc. "Google Maps." Web. 28 May 2014. <<https://maps.google.com/>>.

Handbook on Satellite Communications. 3rd ed. New York, NY: Wiley-Interscience, 2002. Print.

Handbook on Satellite Communications: Fixed-satellite Service. Geneva: International Telecommunication Union, 1985. Print.

Ippolito, Louis J. *Satellite Communications Systems Engineering: Atmospheric Effects, Satellite Link Design, and System Performance*. Chichester, West Sussex, England: Wiley, 2008. Print.

Logsdon, Tom. *The Navstar Global Positioning System*. New York: Van Nostrand Reinhold, 1992. Print.

Michael Russell Rip, James M. Hasik. "The Precision Revolution: GPS and the Future of Aerial Warfare". Naval Institute Press. 2002.

Moore, Dr. R. "A Glossary of British Nuclear Weapons." *Journal of BROHP* (2004). Print.

NMEA 0183--Standard for Interfacing Marine Electronic Devices. Version 3.01. ed. New Bern, North Carolina: NMEA, 2002. Print.

Noble, Joshua, "Programming Interactivity: A Designer's Guide to Processing". Arduino and open Framework. O'Reilly Media. 2009

Nugent, Rory. "Building an Arduino on a Breadboard." *Arduino*. 23 Oct. 2008. Web. 10 Sept. 2014. <<http://arduino.cc/en/pmwiki.php?n=Main/Standalone#.Uzloz-V5Z6c>>.

Pixelis. "Pixelis Maps · Buscador De Coordenadas GPS." Web. 21 May 2014. <<http://www.maps.pixelis.es/>>.

Romo Fuentes, Carlos, and Alfonso Rojas Domínguez. *Diseño De Un Sistema De Control De Navegación Autónoma Para Un Vehículo Marítimo Basado En El Sistema De Posicionamiento Global GPS Utilizando El Microprocesador M68HC111*. México D.F.: Facultad De Ingeniería UNAM, 2002. Print.

Sklar, Bernard. *Digital Communications: Fundamentals and Applications*. Englewood Cliffs, N.J.: Prentice-Hall, 1988. Print.

SparkFun. "GPS Recievers." *GPS Category*. Web. 4 Nov. 2013. <<https://www.sparkfun.com/categories/4>>.

Tetley, L., and D. M. Calcutt. *Electronic Navigation Systems*. 3rd ed. Oxford: Butterworth-Heinemann, 2001. Print.

The National Coordination Office for Space-Based Positioning, Navigation, and Timing. "The Global Positioning System." *GPS System.gov*. Web. 11 Apr. 2014. <<http://www.gps.gov/>>.

Valls, Oriol, and Jaroslav Roztocil. *GPS Time Synchronization through IEEE-1588 Protocol*. 2011. Print.