



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Sistema de navegación automática
basado en el Sistema de
Posicionamiento Global (GPS)**

TESIS

Que para obtener el título de
Ingeniero en Computación

P R E S E N T A

Jesús Bruno Barba Vilchis

DIRECTOR DE TESIS

M.I. César L. Ordóñez Romero



Ciudad Universitaria, Cd. Mx., 2004

PROLOGO 1

Prologo.

El origen de la navegación parece haber ocurrido desde los principios de la era humana, la invención de la brújula se le atribuye a los chinos y fue utilizada en las guerras para navegar durante climas con mucha niebla. Hoy en día hay diferentes técnicas tanto para la navegación terrestre, como para la aérea, y la marítima.

El concepto de la navegación Satelital fue concebido a principios de la década de los 70's, en un principio se experimento con tres sistemas diferentes antes del GPS: el U.S. Navy Navigation Satellite System (*Transit*), el U.S. Navy's Timation (TIME navigATION) y el U.S. Air Force proyect 621B. Estos 3 sistemas derivaron en el Sistema GPS desarrollado a partir de 1973.

Entre las principales aplicaciones del posicionamiento asistido por satélite se encuentran, la navegación con visibilidad nula (aérea, marítima y terrestre), los sistemas de rastreo y la navegación automática.

La navegación automática surge de la necesidad de realizar excursiones a lugares en los que es sumamente riesgoso el acceso humano, o implica un costo muy elevado para una investigación. Por lo que son necesarios vehículos que naveguen de manera automática es decir que no necesiten de control o de asistencia para su movimiento.

Por esta razón se planteó la investigación y el desarrollo de un sistema capaz de generar sus propias señales de control a partir del sistema de navegación o sistema de posicionamiento global GPS, debido a que este sistema esta disponible para cualquier persona y en prácticamente la totalidad de la superficie y la atmósfera terrestre.

Este sistema debe ser capaz de navegar ya sea en espacios de 2 y de 3 dimensiones además de contar con una interfaz que haga sencilla su implementación.

Este trabajo de tesis intenta cumplir este compromiso revisando los aspectos concernientes a los sistemas de posicionamiento global basados en la navegación satelital, los dispositivos que se utilizan para tener acceso a estos así como las etapas de control y potencia que permitirían el desarrollo del sistema antes propuesto, desde su parte teórica hasta la electrónica empleada para su implementación.

Los temas mencionados en el párrafo anterior serán tratados de manera independiente, poniendo énfasis en el algoritmo de navegación desarrollado, ya que es la parte medular de esta investigación y por ultimo se presenta en los anexos el programa en ensamblador ASM desarrollado para le control.

1. SISTEMA DE POSICIONAMIENTO GLOBAL GPS.....	1
1.1 DEFINICION.....	1
1.2 PRINCIPIOS BÁSICOS DEL FUNCIONAMIENTO DEL SISTEMA GPS.	1
1.3 LOS SISTEMAS DE POSICIONAMIENTO GLOBAL ACTUALES.....	3
1.3.1 NAVSTAR-GPS.....	3
1.3.1.1 Segmento Espacial.....	4
1.3.1.2 Segmento de Control.....	5
1.3.1.3 Segmento de Usuario.....	6
1.3.2 GLONASS (GLOBAL NAVIGATION SATELLITE SYSTEM)	9
1.4 REFERENCIAS:	14

1. Sistema de Posicionamiento Global GPS.

1.1 DEFINICION.

El sistema de posicionamiento global (GPS)ⁱ es un servicio de ubicación que permite conocer con una gran precisión la posición de un usuario que se encuentre en cualquier punto sobre la tierra con la limitante de estar al aire libre.

1.2 Principios básicos del funcionamiento del sistema GPS.

El sistema de posicionamiento global permite a sus usuarios saber de manera muy precisa cual es su ubicación en longitud, latitud y altitud así como su velocidad relativa, es muy útil en la navegación y aplicaciones geodésicas. Las señales de los satélites GPS son captadas por medio de un receptor especial, el cual calcula las distancias a los diferentes satélites y recibe las coordenadas de la ubicación de los mismos, con esta información el receptor calcula su posición.

El sistema GPS esta constituido por una constelación de satélites, cada uno de ellos dotado con relojes atómicos, computadoras, emisores y receptores de radiofrecuencia y por estaciones terrenas que monitorean constantemente a cada uno de los satélites.

Básicamente su funcionamiento es de la siguiente manera:

Cada satélite transmite su posición y el tiempo de su reloj exacto constantemente hacia la tierra, donde un receptor computarizado puede calcular a qué distancia se encuentra de un

ⁱGPS *Global Positioning System*

satélite en particular, multiplicando la velocidad de la luz por el tiempo que tarda en llegar la señal del satélite al receptor GPS. Al combinar las señales de varios satélites, el receptor puede establecer con exactitud su propia posición e inclusive su velocidad.

La idea básica de la determinación de la posición se basa en la triangulación de los satélites. Para triangular un receptor GPS calcula la distancia en base al tiempo de travesía de la señal a través de las capas de la atmósfera, conociendo de antemano la velocidad de la luz. Para calcular el tiempo de travesía, los receptores GPS necesitan calcular los tiempos en ambos relojes, el del receptor y el del satélite, de una manera muy precisa.

Los sistemas GPS constan de una constelación de satélites que orbitan la tierra, el número y altitud varían de acuerdo al sistema del que se trate. Existen dos sistemasⁱⁱ de posicionamiento global, NAVSTARⁱⁱⁱ y GLONASS^{iv}, las 24 horas del día los 365 días del año estos satélites transmiten señales de radiofrecuencia que contienen información de tiempo y posición, así como datos de las efemérides y el almanaque de los satélites. Esto permite que cualquiera que tenga acceso a un receptor GPS pueda determinar su posición exacta sobre la tierra. Las señales del GPS están disponibles para un número ilimitado de usuarios simultáneamente.

Las efemérides son un conjunto de parámetros utilizados por un receptor GPS para calcular la localización de un satélite GPS y el comportamiento de su reloj. Las efemérides contienen la posición exacta del satélite que la envía así como también la

ⁱⁱ Existe el proyecto para un tercer sistema llamado Galileo, operado por la Unión Europea

ⁱⁱⁱ NAVSTAR pertenece a los Estados Unidos de Norteamérica

^{iv} GLONASS pertenece a la Federación Rusa

información de su reloj. Los datos de efemérides se transmiten por cada satélite cada 30 segundos.

El almanaque es un conjunto de parámetros utilizados por un receptor GPS para calcular la localización aproximada de un satélite GPS y la diferencia en tiempo del reloj del satélite respecto a su reloj interno. EL almanaque contiene datos mas completos que los contenidos en las efemérides, tales como los datos de concernientes a su orbita, reloj y el estado del satélite Cada satélite GPS contiene y transmite los datos del almanaque de todos los satélites.

1.3 Los sistemas de posicionamiento global actuales

1.3.1 NAVSTAR-GPS.

El sistema NAVSTAR-GPS consta de una constelación de 24 satélites que orbitan la tierra dos veces por día a una altitud de aproximadamente 20,000 Km. [NASA 2002]

El sistema NAVSTAR fue desarrollado en la década de los 70s, y perfeccionado en la década de los 80s, por el departamento de defensa de Estados Unidos, con el fin de proveer información de posición y navegación de manera global y de forma ininterrumpida a la milicia de ese país. Así mismo, con paso del tiempo se han desarrollado un gran número de aplicaciones para el GPS en el ámbito civil. La información de posicionamiento y navegación que ofrece el GPS es útil en aplicaciones de transportación terrestre, aviación civil, comercio marítimo, topografía, construcción, minería, agricultura, ciencias geológicas, energía eléctrica, telecomunicaciones, y actividades recreacionales a la

intemperie. La tecnología GPS utiliza el procesamiento de información de tiempo de una forma precisa como base para el cálculo de la posición. Usando relojes atómicos (con una precisión de alrededor de un segundo por cada 16,000,000,000 años), [NASA 2002] cada satélite transmite continuamente el tiempo u hora actual además de su posición.

El Sistema de Posicionamiento Global, se divide en tres segmentos: segmento espacial que se refiere a la constelación de satélites; segmento de control que monitorea, controla y corrige todo el sistema; y el segmento del usuario que consiste en los distintos tipos de receptores. A continuación se da una breve descripción de cada uno de estos segmentos.

1.3.1.1 Segmento Espacial

La constelación satelital del sistema NAVSTAR ha sufrido variaciones con el tiempo. Por ejemplo los primeros satélites GPS en órbita tenían una órbita con inclinación de 63° con respecto al Ecuador. La constelación actual de satélites NAVSTAR consta de 21 satélites principales más tres satélites inactivos de respaldo orbitando la tierra en órbitas casi circulares con una elevación de aproximadamente 20,000 Km. sobre el nivel del mar y con un periodo de 12 horas. Las órbitas de estos satélites tienen una inclinación de 55° con respecto al ecuador y están colocados en seis planos equidistantes entre si con 4 satélites en cada órbita. La separación de los planos de las órbitas es de 60° en ascensión recta. [Leick, 1995] figura 1.1.

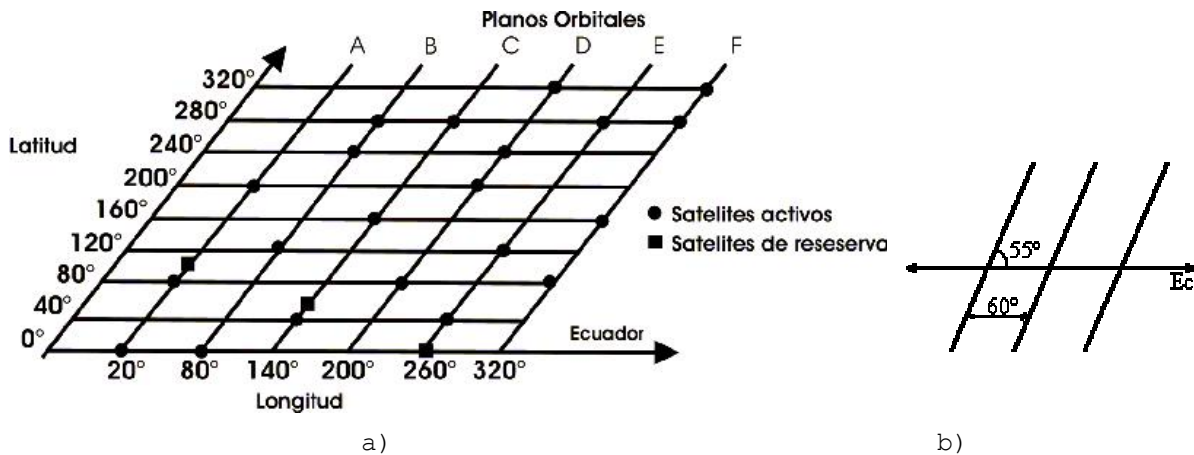


Figura 1.1 a) representación simplificada de la constelación NAVSTAR
 b) separación orbital e inclinación de las orbitas respecto el Ecuador

El Sistema NAVSTAR de Posicionamiento Global se constituyo de 24 satélites para asegurar que por lo menos 5 satélites sean visibles o se encuentren en el horizonte desde cualquier punto sobre la tierra en todo momento. Por lo general siempre se encuentran ocho o más satélites visibles para un receptor GPS. Este hecho es llamado Capacidad de Operación Inicial (COI) [Leick, 1995]

1.3.1.2 Segmento de Control

El segmento de control del sistema NAVSTAR consta de una red de estaciones distribuidas en puntos estratégicos ubicados por toda la tierra, para monitorear controlar y actualizar el sistema de los satélites. Existe una estación de control maestra en el Centro de Operaciones Consolidadas del Espacio, en Colorado Springs. En esta estación se concentra y analiza la información recopilada por las estaciones de monitoreo y con estos datos se

calculan las correcciones en las órbitas de los satélites y correcciones a los relojes atómicos de los satélites. Las estaciones de monitoreo son cinco y se encuentran localizadas en Hawaii, Colorado Springs, Isla Ascención en el Océano Atlántico Sur, Diego García en el Mar Indico Y Kwajalein en el Océano Pacífico Norte. Estas estaciones están equipadas con relojes de cesio y receptores del código P que constantemente monitorean todos los satélites sobre el horizonte.

El monitoreo de estas estaciones es usado para la determinación de las efemérides transmitidas y las simulaciones de reloj. Las correcciones a las órbitas y relojes son retroalimentadas a los satélites mediante las estaciones de control terrestres. Las cuales se encuentran ubicadas en Ascención, Diego García y Kwajalein.

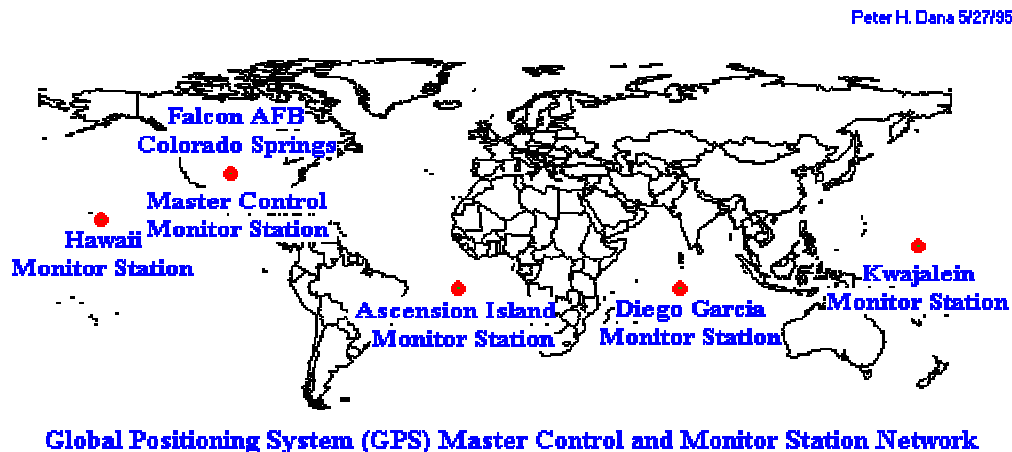


Figura 1.2 estaciones de control del sistema NAVSTAR

1.3.1.3 Segmento de Usuario

El segmento del usuario está integrado por los receptores GPS de los cuales existen gran variedad en el mercado, orientados a los distintos usuarios del sistema GPS. Ya que la utilidad del sistema, han derivado en un sin fin de aplicaciones, además de

las relacionadas con la ubicación por si sola, es necesario diseñar y desarrollar equipos con diferentes características para adaptarse a las nuevas y distintas necesidades de los usuarios.

Los receptores GPS determinan su posición a partir de recibir las señales de radiofrecuencia emitidas por los satélites para que estos puedan calcular qué tan lejos se encuentran de cada satélite. Debido a que los satélites están orbitando a una distancia aproximada de 20,000 Km. Sobre de la superficie terrestre, las señales de radiofrecuencia llegan débiles al receptor, (por lo que entre el receptor y los satélites debe haber el menor numero de obstáculos para poder funcionar con una buena exactitud).

La idea original por lo que se desarrollo el sistema *NAVSTAR-GPS*, que era usarlo como una herramienta que proveyera información precisa sobre la ubicación como ayuda en la navegación de la unidades militares estadounidenses, es utilizada hoy en día con gran éxito. Esto es, conocer la posición del receptor en cualquier momento del día dentro de un sistema de referencia creado para tal fin. Esto se conoce como posicionamiento absoluto. Originalmente se planeo y funciona con dos niveles de servicio GPS: uno para acceso civil y otro para uso exclusivamente militar utilizando un código encriptado. Inicialmente las señales GPS de uso civil estaban sujetas a un proceso conocido como Disponibilidad Selectiva o *Selective Availability (SA)* que consistía en la introducción intencionada de interferencia (errores de tipo aleatorio) para reducir la precisión de la información transmitida por esa señal y mantenerla dentro de un rango de precisión de aproximadamente 100 metros esto para evitar que el sistema fuese utilizado por fuerzas enemigas o terroristas en contra de los E.U. En mayo del

año 2000, el gobierno de Estados Unidos deshabilitó la modalidad de Disponibilidad Selectiva de las señales GPS, lo que significó una mayor precisión, aproximadamente diez veces mayor. De este modo, por lo general la posición de un receptor puede fijarse actualmente en un radio de menos de 10 metros. Los usuarios del sistema GPS pueden incrementar aún más la precisión de los datos mediante el uso de los siguientes métodos llamado GPS Diferencial (DGPS), el *Wide Area Augmentation System (WAAS)*, el sistema europeo EGNOS o el *Multifuncional Transport Satellite-based Augmentation System (MSAS)*. Mediante estos sistemas es posible calcular señales de error de los satélites GPS y posteriormente corregir los datos de posición.

Gracias a las correcciones calculadas a partir de los sistemas WAAS, EGNOS y MSAS, es posible lograr precisión en un rango de 3(m). Usando DGPS se alcanza precisión en tiempo real dentro de un radio menor a 10 metros^v. Con el uso de DGPS y un posterior tratamiento estadístico de los datos, pueden alcanzarse rangos de precisión menores a un metro en posicionamiento fijo (esto es sin movimiento). De esta forma, se puede confiar en el GPS como una forma extremadamente precisa de obtener datos de posicionamiento que pueden ser aplicados junto con los datos geográficos acumulados a través de los años. Los geógrafos a lo largo de la historia han cartografiado prácticamente toda la tierra, por lo que se pueden encontrarse mapas con muy distintos niveles de detalle dependiendo de lo que se requiera.

^v La precisión del sistema GPS depende de la tecnología utilizada en cada receptor. En la mayoría de la bibliografía consultada en esta investigación, el dato oscila entre 8 y 10 (m).

1.3.2 GLONASS (GLOBAL NAVIGATION SATELLITE SYSTEM)

El sistema GLONASS es un sistema de navegación por satélite similar al NAVSTAR pero con importantes diferencias. El sistema está administrado por las Fuerzas Espaciales Rusas para el Gobierno de la Confederación Rusa y tiene importantes aplicaciones civiles además de las militares.

Al igual que en el sistema NAVSTAR, existen dos señales de navegación : la señal de navegación de precisión estándar (SP) y la señal de navegación de alta precisión (HP). La primera está disponible para todos los usuarios tanto civiles como militares que deseen emplearla en todo el mundo, y permite obtener posiciones con errores entre 57 y 70 metros sobre la superficie (99.7% de probabilidad), y con errores en la posición de altura de 70 metros (99.7% de probabilidad), las componentes del vector velocidad con precisión de 15 cm/s (99.7% de probabilidad) y el tiempo con precisión 1 milisegundo (99.7% de probabilidad). Estas características pueden ser mejoradas empleando sistemas diferenciales similares a los empleados con NAVSTAR.[GLONASS 2002]

La constelación completa está formada por 21 satélites activos y 3 de reserva situados en tres planos orbitales separados 120°. Esto permite que sobre el 97% de la superficie terrestre sean visibles al menos 4 satélites de forma continua, frente a los 5 satélites (mínimo) que pueden ser vistos en el 99% de la superficie terrestre si la constelación es de 24 satélites (GPS). De los 24 satélites de la constelación GLONASS, periódicamente se seleccionan los 21 que resultan dar la combinación más eficiente y los 3 restantes son dejados en

reserva. Se ha planteado la posibilidad de aumentar la constelación a 27 satélites, de los cuales 24 estarían activos. Si uno de los 21 satélites operativos se avería, el sistema baja al 94.7% su probabilidad de éxito. El sistema de mantenimiento de la constelación prevé la activación de uno de los satélites en reserva o el lanzamiento de 3 satélites para sustituir a los averiados o ser dejados en reserva para usos futuros.

La primera nave tipo GLONASS (el COSMOS 1413) fue lanzado el 12 de Octubre de 1982, y el sistema GLONASS comenzó a operar oficialmente el 24 de Septiembre de 1993 por decreto del Presidente de la Federación Rusa, Boris Yeltsin.

Cada satélite GLONASS describe una órbita circular a 19100 Km respecto a la superficie terrestre con una inclinación de 64.8° . El periodo orbital es de 11 horas y 15 minutos. La configuración del sistema GLONASS proporciona datos de navegación a usuarios que se encuentren incluso por encima de los 2000 Km. sobre la superficie terrestre. [GLONASS 2002]

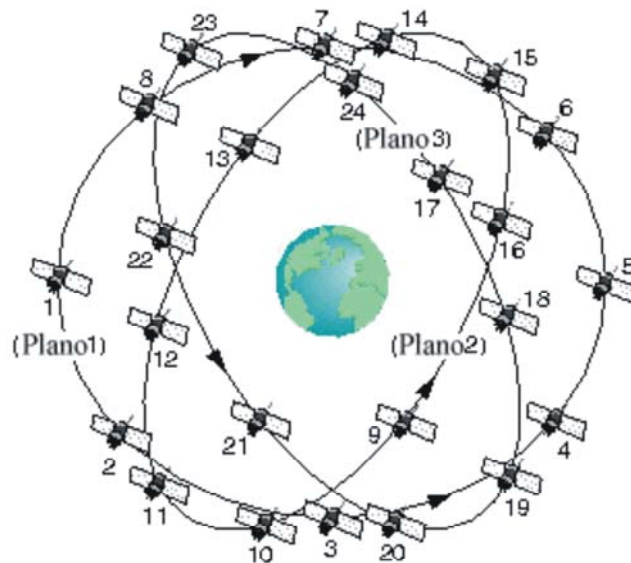


Figura 1.3 constelación GLONASS en los 3 diferentes planos

Para realizar posicionamiento en 3D, medir velocidades y realizar referencias de tiempo, el sistema GLONASS emplea señales radiofrecuencia transmitidas de forma continua por satélites.

El receptor GLONASS recibe señales de navegación de al menos cuatro satélites y mide sus pseudodistancias y velocidades. Simultáneamente selecciona y procesa el mensaje de navegación incluido en la señal de navegación. El receptor procesa toda la información recibida y calcula las tres coordenadas de posición, las tres componentes del vector velocidad y el tiempo.

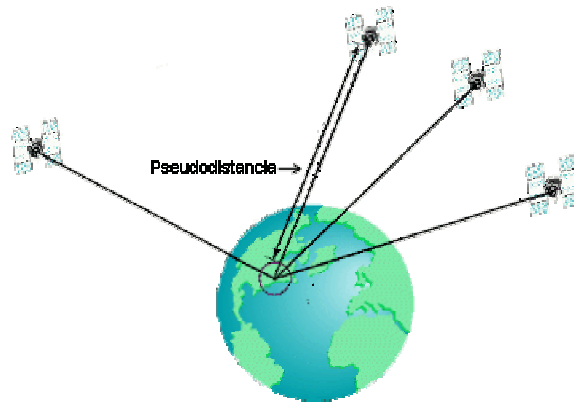


Figura 1.4 Distancia de los satélites a la tierra

Todos los satélites del sistema GLONASS están equipados con relojes de cesio que son corregidos dos veces al día, lo que permite una precisión de 15 nanosegundos en la sincronización de tiempos de los satélites respecto al Sistema de Tiempos GLONASST. El Sistema de Tiempos GLONASS (GLONASST) es generado en la Central de Sincronización de Tiempos de Moscú, cuyos relojes de hidrógeno tiene una variación diaria inferior a 5×10^{-14} . El sistema GLONASS proporciona referencias de tiempo en el Sistema

de Tiempos GLONASS, mantenido en Moscú y en el sistema UTC (CIS), mantenido en el Centro Meteorológico Principal del Servicio Ruso de Tiempos y Frecuencias de Mendeleev (VNIIFTRI). La introducción del UTC en lugar del TAI (Tiempo Atómico Internacional) se debe a que el TAI es un sistema de tiempo continuo que no tiene en cuenta la disminución de la velocidad de la tierra respecto al Sol, que en la actualidad implica 1 segundo al año, lo que significaría un problema para la sincronización con el día solar. El tiempo NVSATAR (GPST) no es incrementado un segundo cada año, por lo que la diferencia de tiempos GLONASS y GPS no es igual cada año. .[GLONASS 2002]

A finales de los años 70 comenzó en Rusia la investigación en el campo del sistema GLONASS diferencial, lo que significa que esta investigación comenzó al tiempo que se desarrollaba el sistema GLONASS. Los científicos del Instituto Central de Investigación de las Fuerzas Espaciales Rusas (TsNII VKS), el Instituto Ruso de Investigación de Ingeniería de Vehículos Espaciales (RNII KP) y la Corporación Científica de Producción de Mecanismos Aplicados (NPO PM) tomaron parte activa en esta investigación.

Pero, debido a diversas causas, la implementación del sistema GLONASS diferencial en Rusia no llegó a su fin. La falta de disponibilidad selectiva en el GLONASS fue decisiva para que esto ocurriera, ya que la precisión estandar del sistema resultaba suficiente para los usuarios rusos.

En 1990-91 los trabajos en este campo volvieron a revitalizarse debido a extensión del sistema DGPS incluso en territorio ruso y a que ciertas compañías extranjeras mostraron gran interés en introducirse en el mercado ruso de equipamiento.

Bajo estas circunstancias, el interés de los usuarios rusos y de los fabricantes de equipos diferenciales aumentó considerablemente y los trabajos para la creación de estaciones diferenciales para diversas aplicaciones se aceleraron.

Actualmente está en proyecto la creación de sistemas diferenciales de área local (LADS)^{vi} y de área regional (RADS)^{vii} para el control del tráfico aéreo y marítimo, pero debido a la necesidad de canales específicos para la transmisión de las correcciones diferenciales, su uso por parte de usuarios particulares es problemático.

En Rusia existe la tendencia a crear una red de sistemas diferenciales departamentales orientada a usuarios específicos. Estos sistemas son los LADS pero sus zonas operativas no cubren la totalidad del territorio ruso. Una posible solución sería incrementar el número de LADS para dar servicio a la totalidad del territorio, pero resulta demasiado costoso. Por este motivo existen propuestas para emplear otro tipo de sistemas diferenciales.

En 1994, el Instituto Central de Investigación de las Fuerzas Espaciales Rusas junto con el Centro de Coordinación de Información Científica de las Fuerzas Espaciales Rusas (KNITs VKS) llevaron a cabo el proyecto del futuro sistema diferencial ruso en el que se emplearían las infraestructuras de las bases de tierra del Complejo Ruso de Control de Vehículos Espaciales. Este sistema diferencial sería capaz de dar servicio a prácticamente totalidad de usuarios en Rusia.

^{vi} LADS Local Area Diferencial Sistem

^{vii} RADS Regional Area Diferencial Sistem

1.4 Referencias:

[Leick, Alfred, GPS Satellite Surveying]

[Bao-Yen, James; "Fundamentals of Global Positioning System Receivers, A Software Approach" 2000]

[Grewal Mohinder S. "Global Positioning Systems, Inertial Navigation, and Integration"]

[<http://www.nasa.gov/>]

[<http://www.glonass-center.ru/>]

[[http://europa.eu.int/comm/dgs/energy transport/galileo/programme/architecture en.htm](http://europa.eu.int/comm/dgs/energy_transport/galileo/programme/architecture_en.htm)]

2	FUNCIONAMIENTO Y APLICACIONES DE GPS .	1
2.1	FUNCIONAMIENTO DEL SISTEMA GPS	1
2.1.1	EL MÉTODO DE TRIANGULACIÓN.	1
2.1.2	TRIANGULACIÓN APLICADA EN LOS SISTEMAS GPS	4
2.1.3	CARACTERISTICAS DE LA SEÑAL GPS	10
2.1.3.1	CARACTERISTICAS DE LA SEÑAL NAVSTAR-GPS	10
2.1.3.2	CARACTERISTICAS DE LA SEÑAL GLONASS	12
2.3.1	EL RECEPTOR GPS	14
2.3	APLICACIONES DEL SISTEAMA GPS	21
2.3.1	APLICACIONES EN LA INVESTIGACIÓN	21
2.3.2	APLICACIONES DE TRANSPORTE	22
2.3.3	APLICACIONES EN LA INDUSTRIA	23
2.4	REFERENCIAS :	24

2 Funcionamiento y Aplicaciones de GPS.

2.1 Funcionamiento del Sistema GPS

Los sistemas GPS basan su funcionamiento en la medición de distancias entre el equipo receptor y los satélites de la constelación GPS. Para realizar los cálculos de su posición el receptor necesita conocer su posición relativa respecto de cada satélite, es decir, la posición de cada satélite y la distancia que hay entre este y el receptor. Partiendo de esta información, el receptor aplica el concepto de triangulación y de esta forma calcula su posición.

2.1.1 El Método de Triangulación.

El método de la triangulación hace posible encontrar la posición de un punto en el espacio a partir de distancias medidas desde ese punto a un número determinado de posiciones conocidas. Este método puede ser aplicado en espacios de una dimensión (Recta) a partir de dos puntos, de dos dimensiones (el plano) donde se necesitan tres posiciones conocidas para determinar la posición desconocida (de ahí el nombre de triangulación), este mismo principio puede usarse en espacios de tres dimensiones. Por ejemplo, en el caso de una dimensión, en la que se busca determinar la posición de un punto \mathbf{R} sobre una recta. Se conocen las distancias \mathbf{a}_1 y \mathbf{a}_2 , medidas sobre la recta desde \mathbf{R} hasta dos puntos j_1 y j_2 respectivamente figura 2.1

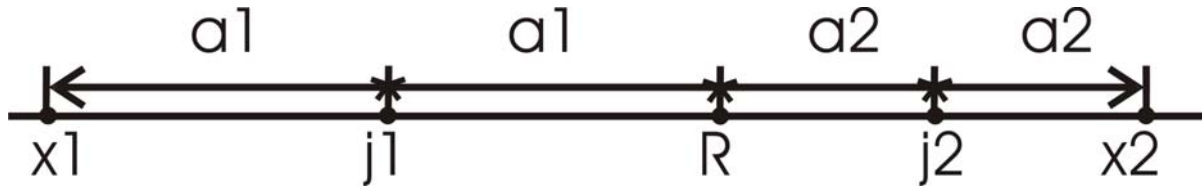


Figura 2.1 Triangulación en un Espacio de una Dimensión

Desde el punto j_1 , la distancia a_1 define dos puntos sobre la recta, x_1 y R . A su vez, desde j_2 la distancia a_2 define otro par de puntos, x_2 y R . Si las distancias a_1 y a_2 han sido medidas correctamente (caso ideal), uno de los puntos medidos desde j_2 coincidirá con uno de los puntos medidos desde j_1 . El punto de coincidencia es, lógicamente, la posición buscada R .

Para dos dimensiones se requiere conocer tres posiciones y a su vez la medida de tres distancias entre éstas posiciones y el punto buscado. Cada distancia forma una circunferencia de posibles posiciones donde puede encontrarse el punto buscado. El punto en donde se intersecten las tres circunferencias es la posición del punto buscado. Figura 2.2

Si las distancias medidas desde el punto buscado hasta las posiciones de referencia (de las cuales se conoce su posición) carecen de precisión, las circunferencias generadas no se intersectarán en un punto y la posición buscada no quedará completamente definida, en su lugar se obtendrá un área en la que uno de los puntos de esa área es el buscado (figura 2.2 b). Si el error que ocurrió en la medición de las distancias afectó por igual a cada una de ellas, puede aplicarse un método iterativo de aproximaciones para reducir el área de posibilidades y obtener una solución o punto. Si se conoce un número mayor de posiciones de referencia y distancias medidas del punto buscado a esos puntos, el método de aproximaciones convergerá más rápido. La geometría que presenten los puntos de referencia (la posición

relativa donde se encuentran entre ellos) también influye en la determinación de la solución, al tener una buena geometría, a pesar de existir error en la medición de distancias, el resultado será un área de solución menor, donde obtener el punto buscado será más fácil de determinar.

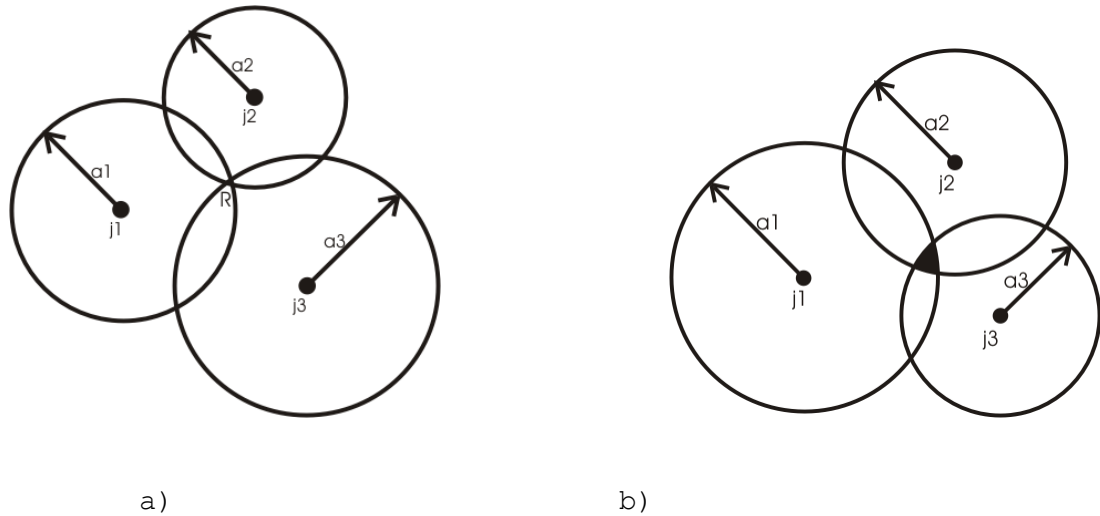


Figura 2.2 a) Triangulación en Dos Dimensiones. a) Posición Unívocamente Determinada.
 b) Posición no determinada unívocamente debido a la imprecisión en la Medida de las Distancias de Referencia.

Para el caso de tres dimensiones, como puede suponerse, es necesario conocer cuatro posiciones y sus distancias correspondientes, medidas entre estas y el punto cuya posición se desea encontrar para llegar a una solución por triangulación. Cada punto de referencia dará origen a una esfera de posibles posiciones del punto buscado. Dos esferas de este tipo se intersectan en una circunferencia. Con tres esferas se obtienen dos posibles puntos de solución. La cuarta esfera determina unívocamente la posición buscada.

Así mismo, si las distancias medidas desde cada punto de referencia carecen de precisión, la solución no será un punto, en su lugar obtendrá un conjunto de ellos. La posición del punto buscado deberá aproximarse hasta alcanzar la precisión deseada.

2.1.2 Triangulación aplicada en los sistemas GPS

En los sistemas GPS, la posición del receptor GPS (del usuario) es obtenida a partir de las posiciones de los satélites de la constelación GPS (puntos de referencia conocidos) y las consecuentes distancias medidas entre ellos y el receptor. A partir de estos datos, el receptor aplica el método de triangulación para así calcular su posición. Idealmente se necesitan cuatro puntos de referencia para determinar unívocamente una posición en el espacio de tres dimensiones. Sin embargo, en el caso de los sistema GPS se sabe que la posición del receptor debe encontrarse en algún punto de la superficie terrestre. Por lo tanto, la tierra puede considerarse como la cuarta esferas que se usan en la determinación de la posición y sólo se necesitan otros tres puntos de referencia. figura 2.3.

Por medio de los satélites S1, S2 y S3 en la figura 2.3 la posición del receptor GPS puede restringirse a dos posibles puntos. Sólo uno se encuentra sobre la tierra, el otro se encuentra en algún lugar en el espacio. Matemáticamente, el receptor GPS elimina el punto fuera de la tierra y establece las coordenadas del equipo receptor.

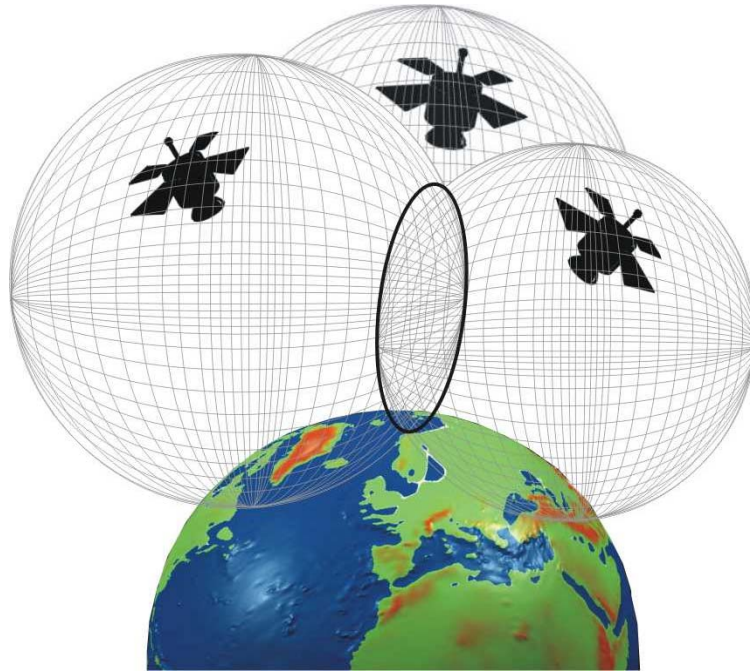


Figura 2.3 Triangulación en 3 dimensiones tomando la tierra como una cuarta esfera

Lo expuesto con anterioridad, se cumple solo si las distancias desde cada satélite hacia el receptor han sido medidas con gran precisión. Dado que en el receptor GPS carece de ecuaciones que se adapten exactamente a la superficie terrestre, este debe estimar la altitud a la que se encuentra en cierto tiempo. Inicialmente el receptor considera su altitud como cero referenciada en el centro de la tierra, por medio de iteraciones da incrementos de altitud hasta coincidir con alguno de los dos posibles puntos de su posición. En caso de que las distancias que hayan sido medidas desde los satélites carezcan de precisión, al equipo receptor le será imposible estimar su altitud. En cuyo caso, el receptor sólo entregará valores de latitud y longitud de su posición, sin arrojar información de altitud. Cuando hay 4 o más satélites visibles para el receptor, este calcula las pseudo-distancias respecto a cada uno de esos satélites y obtendrá un sistema con igual o mayor número de ecuaciones que incógnitas. Esto mejora el cálculo de la posición del aparato receptor,

debido a que tienen más información para calcular la corrección debida al error de medición.

Las distancias entre el receptor y los satélites se obtienen por medio del cálculo realizado sobre el retardo temporal que ocurre entre el instante en que cada satélite envía una señal hasta el momento en que el receptor la recibe. Un receptor GPS mide el tiempo que le toma a la señal de navegación viajar desde el satélite hasta el receptor. Puesto que se conoce la velocidad a la que viaja la señal (ya que es una onda electromagnética y por lo tanto viaja a la velocidad de la luz), puede calcularse qué distancia ha recorrido verificando el tiempo que le tomó llegar de un punto a otro. La distancia entre el receptor y cada satélite es determinada usando la fórmula $d = ct$, donde c corresponde a la velocidad de la luz en el vacío y t es el tiempo que tarda la señal en hacer el recorrido desde el satélite hasta el receptor.

El proceso que realiza un sistema GPS, para calcular la posición de un aparato receptor, se describe a continuación:

Un satélite Inicia la transmisión de un patrón digital como parte de su señal en ciertos intervalos de tiempo. Inicia la transmisión en un tiempo conocido, por ejemplo, exactamente al mediodía.

A continuación el receptor GPS inicia ejecutando el mismo patrón digital que el satélite, también al mediodía. Cuando la señal del satélite es captada por receptor, la transmisión del patrón digital de referencia emitido por el satélite llega con unos bits de retardo con respecto del patrón ejecutado por el receptor, debido al tiempo que transcurrió desde que la señal

salió del satélite hasta que alcanza al receptor. El tiempo de retraso, o el número de bits de diferencia, es igual al tiempo que viajó la señal. El receptor multiplica este tiempo por la velocidad de la luz para determinar la distancia que ha viajado la señal. Si el viaje de la señal ha sido en línea recta, la distancia obtenida del cálculo será la distancia que hay entre el satélite y el equipo receptor.

Mediante el cálculo de la distancia el receptor aplica el método de la triangulación para generar una ecuación de una esfera por cada satélite que haya captado. El punto donde todas estas esferas se intersectan es la posición donde se encuentra el receptor.

Es necesario proveer al sistema con un mecanismo para medir tiempo. Esto sería sencillo si se supiera exactamente en qué momento las señales abandonan los satélites y en qué momento llegan al receptor, la solución de esto es el problema principal para los Sistemas de Posicionamiento Global. Una solución consistiría en colocar relojes extremadamente precisos y perfectamente sincronizados en ambos dispositivos, es decir, tanto en los receptores como en los satélites.

Ya que no se puede contar con un reloj perfecto, los relojes en los receptores y en los satélites tienen un error que afecta las mediciones de distancia. El sistema GPS evita el error de los satélites haciéndolo mínimo al utilizar relojes atómicos y correcciones regulares a los mismos relojes. Pero, para que todos los relojes estuviesen en perfecta sincronía serían necesarios relojes atómicos no solamente en los satélites, sino también en los mismos receptores, esto se debe a que las medidas de los tiempos varían en el orden de nanosegundos (ns), si bien

los satélites cumplen esta condición, esto no es posible para los receptores, ya que el incorporar un reloj atómico aumentaría su precio, siendo este incosteable para los usuarios.

Los Sistemas GPS proporcionan una solución efectiva para este problema, tomando en cuenta que el receptor cuenta con un reloj de cuarzo que es preciso aunque no lo suficiente como un reloj atómico. El receptor procesa todas las señales que recibe de los satélites y de manera matemática encuentra su posición y el tiempo exacto que tomó cada señal en viajar simultáneamente. Al hacer esto, introduce una variable más por determinar, el error de tiempo en el reloj del receptor, denotado **BUT (Bias Clock User error in Time)**; este es el error producido como consecuencia del desfase entre el reloj del receptor y los relojes de cada uno de los satélites.

De esta manera, la solución al problema es la introducción de esta nueva variable al sistema (sumada a las tres incógnitas de las coordenadas del receptor) generada por la falta de sincronía que existe entre el reloj de los satélites y el reloj del receptor. Al agregar esta variable en total, el receptor debe calcular cuatro variables: latitud, longitud, altitud y el error de tiempo para el reloj del receptor GPS.

Ya que el equipo receptor efectúa todos los cálculos de tiempo así como distancias, haciendo uso de su reloj interno, todas las distancias calculadas serán proporcionalmente incorrectas; Por lo que el receptor calcula un ajuste de distancias que permita que las cuatro esferas, cuyos centros son las posiciones en las que se encuentran los satélites, puedan intersectarse en un punto perteneciente a la superficie terrestre, realizando un ajuste al valor de altitud calculado, cabe mencionar que en algunos casos

puede tomar puntos situados dentro de la atmósfera terrestre. De esta forma, para fijar una posición lo suficientemente precisa es necesario contar con 4 satélites visibles al receptor como mínimo y no 3 como la teoría lo indica en un principio. De hecho, si los relojes del receptor GPS y de los satélites se encontrasen en perfecta sincronía, la posición calculada respecto a cualesquiera tres satélites de cuatro a la vista debería ser exactamente la misma, sin importar la combinación.

Debido a que el intervalo de tiempo es calculado a partir de relojes no precisos cuyos errores deben corregirse, se usa el término de 'pseudo-distancias' para hacer referencia a las distancias medidas por el receptor.

También, es necesario indicar que la posición del receptor es determinada a partir de las posiciones en las cuales se encuentran los satélites en órbita. La distancia medida, por lo tanto, también puede ser afectada por las distintas perturbaciones orbitales que sacan a los satélites de sus órbitas teóricas. Pese a estos tipos de error, el error de reloj es el que más afecta al determinar las distancias a los satélites.

Por todo esto la función principal del equipo receptor es captar las señales emitidas por al menos cuatro satélites, para después a partir de la información recibida de cada uno de estos satélites calcular matemáticamente su posición sobre la superficie o la atmósfera terrestre. La información básica que provee un receptor es latitud, longitud y altitud de su posición actual. El sistema GPS, además de la posición, ofrece una referencia de tiempo muy exacta (UTC¹), para así poder sincronizar los relojes locales. Si son visibles más de cuatro satélites, el

¹ Universal Coordinated Time

receptor proporciona la velocidad a la que se está moviendo el receptor. Dependiendo del receptor estos pueden a partir de los datos GPS interpretar información, como mapas, para convertirse en un aparato más útil y sin complicaciones al usuario. Se pueden almacenar mapas en la memoria del receptor o conectarlo a una computadora que posea mapas más detallados.

2.1.3 CARACTERÍSTICAS DE LA SEÑAL GPS

Como se menciona en el capítulo anterior existen 2 sistemas de posicionamiento global NAVSTAR y GLONASS de EU. Y Rusia respectivamente, a continuación se da una breve explicación de la señal que transmiten estos sistemas GPS

2.1.3.1 CARACTERÍSTICAS DE LA SEÑAL NAVSTAR-GPS

Todas las señales GPS se derivan de una frecuencia base f_0 de 10.23 (MHz) que se obtiene de los relojes atómicos que van a bordo de cada satélite. Los satélites transmiten en frecuencia modulada $L_1=1575.42$ (MHz) y $L_2=1227.6$ (MHz) con dos tipos de código y un mensaje de navegación llamado telemetría. Las dos señales portadoras en la banda L (L_1 y L_2) se generan mediante la multiplicación entera de f_0 de la siguiente manera:

$$f_0 = 10.23(\text{MHz}) \quad \text{Portadora } L_1 = 154f_0 = 1575.42(\text{MHz}), \lambda_1 = 19(\text{cm})$$

$$\text{Portadora } L_2 = 120f_0 = 1227.60(\text{MHz}), \lambda_2 = 24.4(\text{cm})$$

Los mensajes enviados por los satélites, hace uso de dos códigos para su adquisición. Estos códigos se caracterizan por tener en ellos un ruido pseudo aleatorio (*Pseudo Random Noise*).

El primer código es llamado código C/A (*Coarse / Acquisition*, Común / Adquisición) que se repite cada milisegundo. El segundo código es llamado código P (Código Preciso) se repite aproximadamente cada 266.4 días. Las señales portadoras L_1 y L_2 son moduladas con el código P mientras que el código C/A se emplea para modular L_1 solamente.

Tabla 2.1 Frecuencias y Portadoras de los Códigos de Modulación de la Señal GPS.

	Código P	Código C/A
Frecuencia	$F_0 = 10.23 \text{ Mhz.}$	$F_0/10 = 1.023 \text{ Mhz.}$
Portadora	L_1 y L_2	L_1

El 31 de Enero de 1994, fue implementado el tan esperado *antispoofing* (AS). El propósito del AS es lograr que el código P esté disponible para usuarios autorizados. Los usuarios deberán estar equipados con un Decoder (dispositivo descriptador o 'llave') para poder tener acceso al código P. El AS se implementa a través de una modificación en la fórmula matemática del código P usando una regla secreta. Al código P encriptado se le llama código P(Y) (o código Y).

El código P(Y) de precisión es el código principal usado para navegación militar. Es un código de ruido pseudo-aleatorio (PRN) generado matemáticamente al mezclar otros dos códigos pseudo-aleatorios. El código P(Y) no se repite durante 37 semanas. De esta manera es posible asignar semanalmente porciones de este código a varios satélites. Como resultado, todos los satélites pueden transmitir con la misma frecuencia portadora, y aun así pueden ser distinguidos entre sí por las secuencias de código exclusivas que transmiten. Como hay menos de 37 satélites GPS en

órbita, algunas secuencias semanales del código P(Y) permanecen sin uso.

El código C/A o código de común adquisición tiene un periodo de 1 mseg, y consiste de 1023 bits. Cada satélite transmite un conjunto diferente de códigos C/A. Con este código es posible distinguir rápidamente las señales recibidas simultáneamente desde varios satélites. Este código es la base del Servicio de Posicionamiento Estándar (SPS)ⁱⁱ por lo que es necesario resaltar que el código C/A es completamente accesible para uso, por cualquier civil.

Ya que los códigos C/A y P(Y) están sincronizados, el código C/A hace posible la adquisición rápida del código P(Y). El mensaje de navegación, también llamado telemetría, esta modulado sobre ambas portadoras L1 y L2 a una velocidad de transmisión de 50 bps. Este mensaje contiene información sobre las efemérides de los satélites, el tiempo GPS, el comportamiento de los relojes y mensajes del estado del sistema.

2.1.3.2 CARACTERISTICAS DE LA SEÑAL GLONASS

Al igual que el sistema NAVSTAR-GPS, cada satélite modula su frecuencia portadora L1 con dos cadenas de secuencias PRN (Código C/A para uso civil y código P para uso militar) y contienen el mensaje de navegación. La portadora L2 es modulada exclusivamente por la suma del código P y el mensaje de navegación. Los códigos C/A y P son iguales para todos los satélites, por lo que no

ⁱⁱ SPS Standard Position Service

permiten la identificación de los satélites como ocurre en NAVSTAR.

Cada satélite de la constelación transmite dos tipos de señal : L_1 de precisión estandar (SP) y L_2 de alta precisión (HP). Se emplea el sistema de múltiple acceso por división en frecuencia. Esto significa que cada satélite transmite señales en su propia frecuencia, lo que permite su identificación.

Si bien estos valores pueden ser modificadas en circunstancias excepcionales. Existen satélites que transmiten a la misma frecuencia, pero están situados en las antípodas, por lo que no pueden ser "vistos" simultáneamente por un usuario.

Código C/A

Se trata de un código PRN (Ruido pseudoaleatorio) generado mediante un registro de desplazamiento de 9 bits, lo que proporciona un longitud de 511 bits. Este código se transmite a 0.511 Mbits/s, por lo que se repite cada 1 ms. Esto produce componentes de frecuencia no deseados a intervalos de 1 KHz que pueden dar lugar a correlaciones cruzadas no deseables entre fuentes de interferencias. No existe posibilidad de correlaciones cruzadas entre señales de distintos satélites debido a que emplean frecuencias distintas (FDMA). Se elige un código corto para permitir la rápida adquisición del código por parte del usuario. La velocidad de transmisión implica que un bit tarda 1.96×10^{-6} segundos en transmitirse, lo que significa 587 metros.

Código P :

Se trata de un código PRN secreto dedicado al uso militar, por lo que la información disponible sobre éste código es poca y

ha sido obtenida tras el análisis realizado por organizaciones independientes. Es un código generado mediante un registro de desplazamiento de 25 bits, por lo que la longitud es de 3355431 bits. Se transmite a 5.11 Mbits/s.

El código P es mucho más largo que el C/A y también producen componentes de frecuencia a intervalos no deseables, pero los problemas de correlaciones cruzadas no son tan importantes como en el código C/A. Evidentemente, al igual que sucede con el código C/A, no existe posibilidad de correlaciones cruzadas entre distintos satélites.

El código P gana en propiedades de correlación respecto al C/A, pero pierde en cuanto a propiedades de adquisición ya que hay 511 millones de cambios de fase posibles. Para solucionar este problema se emplea el código C/A, que es adquirido previamente para reducir el número de posibilidades y permitir de este modo la adquisición del código P de forma más rápida. Esto es posible debido a que el código P se transmite exactamente a una velocidad 10 veces superior a la del código C/A.

2.3.1 EL RECEPTOR GPS

El Receptor GPS es una combinación de Software y Hardware que permiten determinar la posición, velocidad y tiempo de un usuario, además de los parámetros necesarios adicionales que requiera.

Funciones del receptor.

- 1-Identificación y seguimiento de los códigos asociados a cada satélite.
- 2-Determinación de las distancia.
- 3-Decodificación de las señales de datos de navegación para obtener las efemérides, el almanaque....
- 4-Aplicar las correcciones (del reloj, ionosféricas,...).
- 5-Determinación de la posición y velocidad.
- 6-Validación de los resultados obtenidos y almacenamiento en memoria.
- 7-Presentación de la información.

Según la dinámica pueden ser de baja, media y alta dinámica.

Equipos de baja dinámica: Usuarios terrestre. No tiene dinámica angular. Suele ser de un solo canal.

Equipos de media dinámica: Helicópteros y aviones. Son de varios canales.

Equipos de alta dinámica: Helicópteros y aviones. Son múltiples canales.

El concepto "todos los satélites a la vista" implica un seguimiento continuo de 12 satélites, y las nuevas tendencias van dirigidas a la utilización de las constelaciones GPS y GLONASS de manera conjunta, lo que lleva a receptores con 20 canales

El tipo de receptor a utilizar dependerá del tipo de observaciones que el usuario desee, de la disponibilidad de los códigos de modulación de la señal GPS y del costo de cada receptor. Los receptores pueden clasificarse de varias maneras dependiendo de sus características, por ejemplo:

Tabla 2.2 ejemplos de los tipos de receptores GPS dependiendo de sus características

Por su características de Modulación
Código C/A
Código C/A + Portadora L1
Código C/A + Portadora L1 + L2
Código C/A + P + Portadoras L1,L2

Por el tipo de usuario
Receptores Civiles
Receptores militares
Receptores para navegación
Receptores Geodésicos

2.2.1 Elementos esenciales

Un receptor GPS consiste de los siguientes elementos básicos: una antena con pre-amplificador opcional, una sección de radiofrecuencia y de frecuencia intermedia (*RF/IF*), una sección de rastreo y correlación de señal; y un microprocesador que controle el receptor, procese las señales y calcule las coordenadas del receptor. Los receptores también incluyen una fuente de poder y un dispositivo de memoria para almacenar instrucciones y datos. La figura 2.4 muestra el diagrama de bloques de un receptor GPS.

Un sistema primario de navegación debe cumplir con cuatro características fundamentales :

- Precisión
- Continuidad
- Integridad
- Disponibilidad

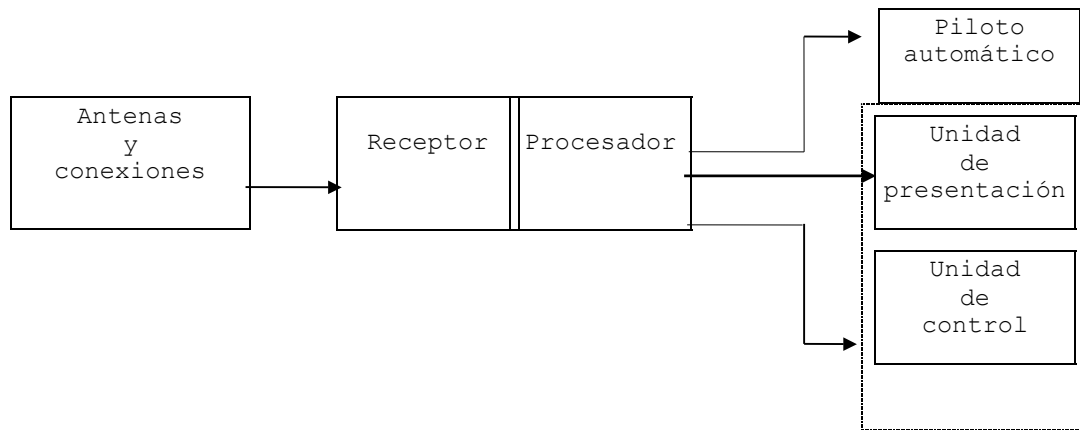


Figura 2.4 Diagrama de Bloques de un Receptor GPS

Antena.- El trabajo de la antena es convertir la energía de las ondas electromagnéticas que llegan de los satélites GPS en una señal eléctrica que pueda manejarse por la electrónica del receptor. El tamaño y forma de la antena son muy importantes, pues de estas características depende, en parte, la habilidad de la antena para recoger y mandar al receptor la débil señal GPS. Una 'antena activa' es aquella que, además del elemento de antena, incluye un preamplificador de bajo ruido (con uno o más filtros asociados) para amplificar las señales. Este tipo de unidad se utiliza cuando la antena se encuentra separada cierta distancia del receptor.

La señal procedente de un satélite llega a la antena del receptor. La señal recibida pasa por un filtro paso banda que elimina las señales que no interesan. A continuación la señal es introducida en un amplificador de gran calidad y con baja ganancia de ruido (4dB) a fin de no arrastrar el ruido por el resto de las etapas. La señal atraviesa un filtro paso bajas que elimina armónicos indeseables y a continuación, en un mezclador, mezclamos la señal recibida (f_c) con la procedente de un oscilador local (f_o) y obtenemos f_c+f_o y f_c-f_o . Mediante un

filtro paso bajas nos quedamos con la frecuencia más baja ($f_c - f_o$). Repitiendo el proceso podemos reducir la frecuencia hasta el nivel adecuado, y entonces amplificaremos la señal y pasaremos a la etapa de procesado. Un esquema puede observarse en la figura siguiente

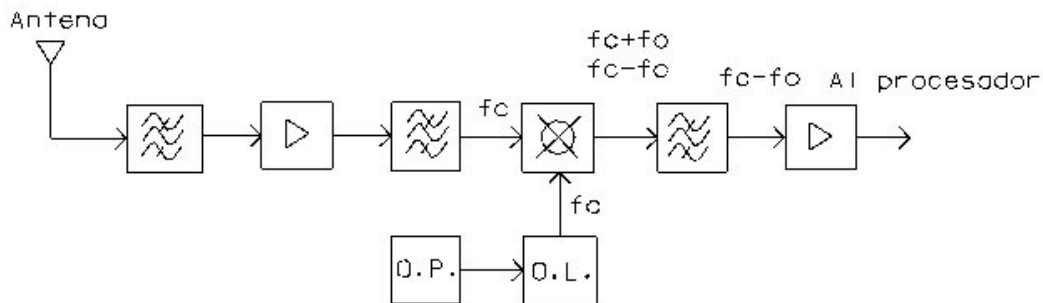


Figura 2.2 Procesador de datos

En el procesador de datos se recibe toda la información anterior y se calcula la posición, velocidad y tiempo. Las funciones básicas que realiza son :

- Controlar el receptor tanto en la etapa de frecuencia intermedia como en la de procesado de las señales
- Decide que satélites seguir (validar, registrar y presentar la información)
- Calcular la pseudo-distancia
- Aplicar las correcciones correspondientes
- Analizar la información para presentarla

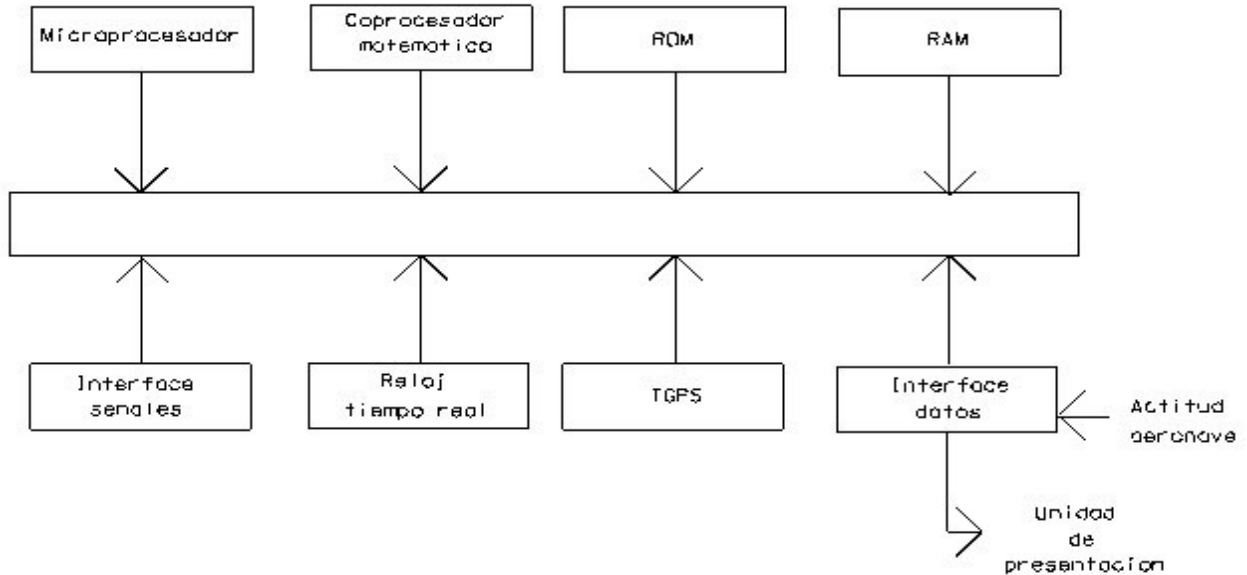


Figura 2.6 Procesador del receptor.

Procesamiento de la señal.- La antena omnidireccional de un receptor GPS recibe simultáneamente las señales de todos los satélites sobre su horizonte. El receptor debe ser capaz de aislar la señal de cada satélite en particular para medir su pseudo-distancia y la fase de su portadora. Para ello, el receptor dispone de varios canales en los que se asigna cada señal. Esto no es difícil puesto que el código único C/A o porción del código P(Y) transmitido por cada satélite permite una fácil discriminación de las distintas señales.

Los canales en un receptor GPS pueden implementarse usando uno de dos modos básicos. Un receptor puede tener canales dedicados que continuamente rastreen satélites particulares. Un mínimo de cuatro de esos canales rastreando la señal LI de cuatro satélites con buena geometría en su posición se requieren para determinar las tres coordenadas de posición y el error o deriva del reloj

del receptor. Canales adicionales permiten el rastreo de más satélites, incluyendo, posiblemente, satélites GLONASS.

La otra forma de canalización se basa en uno o más canales secuenciados. Un canal secuencial "escucha" a un satélite en particular por un período de tiempo realizando mediciones en la señal de ese satélite antes de conmutar a otro, Un receptor con sólo un canal debe conmutar secuencialmente ese canal a por lo menos cuatro satélites para obtener su posición en tres dimensiones. Antes de que una lectura de posición pueda obtenerse, un receptor debe permanecer con la señal de cada satélite durante al menos 30 segundos para adquirir los datos suficientes del mensaje de ese satélite. Una variación del concepto de canales secuenciados es la multiplexión en tiempo de canales. 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 un intervalo de tiempo mucho menor entre lecturas.

Microprocesador y Memoria.- Un microprocesador controla la operación del receptor GPS. Como estos dispositivos se vuelven cada vez 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 dentro del mismo microprocesador o en circuitos integrados auxiliares que son parte del receptor.

El microprocesador del receptor trabaja con muestras digitales de las pseudo-distancias y la fase de portadora, adquiridos por medio de una conversión analógica-digital

realizada en algún punto del flujo de la señal a través del receptor. El receptor usa estas muestras de la información para establecer su posición, velocidad y tiempo. El microprocesador también realiza rutinas de filtrado sobre la información para reducir los efectos del ruido. Además, el microprocesador puede utilizarse para realizar cálculos de conversión de sistemas de coordenadas, para combinar correcciones diferenciales desde bases en tierra o sistemas de aumento basados en satélite. También maneja los comandos de entrada del usuario, el despliegue de información y el flujo de datos a través de su puerto de comunicación, si posee uno. Los receptores más modernos disponen de un microprocesador especializado en Procesamiento Digital de Señales *DSPⁱⁱⁱ* estos son circuitos integrados muy eficientes y pequeños.

2.3 APLICACIONES DEL SISTEMA GPS

En el mundo moderno los sistemas GPS han tomado gran importancia puesto que son de gran ayuda en muy diversas ramas, que pueden ir desde la investigación hasta lo más trivial.

2.3.1 Aplicaciones en la Investigación

Los principales usos de los sistemas GPS en la investigación son el trazado de mapas y levantamiento topográfico. Con la ayuda de un receptor GPS los investigadores toman la posición exacta de ciertos puntos para así poder trazar luego un mapa y de esta forma tener un levantamiento mucho más exacto de la cartografía. En estos mapas que se pueden transformar en bases de datos y servicios de información se muestran todo tipo de detalles tales

ⁱⁱⁱ Digital Signal Processing

como accidentes orográficos, ríos, lagos, carreteras, calles e incluso se pueden encontrar levantamientos mas exactos donde se indica incluso donde se encuentran lugares como cajeros automáticos o restaurantes.

En la arqueología Los sistemas GPS son muy útiles ya que permiten marcar con exactitud los lugares donde se encuentran los hallazgos arqueológicos, esto es de gran ayuda sobre todo cuando los lugares cuentan con una densa vegetación lo que haría casi imposible regresar a un lugar sin la ayuda de los sistemas GPS

2.3.2 Aplicaciones de transporte

Debido a que los sistemas GPS están disponibles en todo el mundo, en el transporte los sistemas GPS han ayudado en varios campos sobre todo en la navegación ya sea en la tierra, el agua (mar o río) o en el aire. En la tierra se utiliza sobre todo para la navegación, los automóviles o camiones pueden con la ayuda de un receptor GPS ubicar su posición exacta y de esta forma poder trazar rutas y/o localizar direcciones. También son utilizados los receptores GPS junto con un sistema de radio transmisión para localizar vehículos ya sea en caso de robo o para evitar que operadores salgan de las rutas establecidas.

En el agua son utilizados receptores GPS en las embarcaciones de todo tamaño para ayudar en la navegación, remplazando en algunos casos instrumentos tradicionales como el sextante, de esta forma los navegantes pueden ubicarse en todo momento incluso con las condiciones climáticas mas adversas y así poder desplazarse de manera segura a su destino.

El lo que respecta a la navegación aérea el GPS ha tomado aún mas importancia ya que prácticamente todos las aeronaves (aviones, helicópteros globos, etc.) incorporan un receptor GPS para de esta forma poder seguir de una manera mas sencilla las cartas de navegación e incluso para poder controlar el avión con el piloto automático.

En la actualidad se encuentra en desarrollo una red mundial de estaciones WAIS para ayudar con su gran exactitud a la navegación aérea y hacer realidad los aviones sin piloto en el ramo de la aviación comercial y el control de trafico aéreo, al conocer la exacta ubicación de las aeronave en todo momento.

2.3.3 Aplicaciones en la industria

En la industria el GPS ayuda a ubicar con exactitud lugares de construcción, rutas de transporte, pozos petroleros, etc. Las compañías mineras utilizan los receptores GPS para la exploración de nuevas minas, además de la administración y el control del equipo y la maquinaria en las minas exteriores tales como canteras o minas de arena, también tienen un papel muy importante en la colocación exacta de explosivos tanto para demolición como para la exploración de dichas minas. En la construcción funcionan para hacer levantamientos topográficos muy exactos para delimitar o medir un terreno o para el trazo de autopistas puentes o carreteras.

Los exploradores utilizan los receptores en excursiones en el bosque, desierto jungla, etc. También son utilizados en carreras de autos, por los ciclistas, cazadores, pescadores, fotógrafos etc. Los sistemas GPS tiene una infinita gama de usos en la industria y el esparcimiento solo limitados por la imaginación

2.4 Referencias:

[Leick, Alfred, "GPS Satellite Surveying" 1995]

[Bao-Yen, James; "Fundamentals of Global Positioning System Receivers, A Software Approach" 2000]

[Grewal Mohinder S. "Global Positioning Systems, Inertial Navigation, and Integration"]

<http://gpshome.ssc.nasa.gov/>

[<http://www.nasa.gov/>]

<http://www.glonass-center.ru/>]

3.	EL PROTOCOLO DE NAVEGACIÓN NMEA 0183.	1
3.1	EL ESTÁNDAR NMEA 0183	1
3.1	INTERFAZ ELECTRÓNICA	2
3.2	FORMATO GENERAL DE LOS MENSAJES	2
3.3	FORMATO DEL MENSAJE NMEA UTILIZADO	4
3.4	REFERENCIAS	9

3. El protocolo de Navegación NMEA 0183.

3.1 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 aplicaciones que utilicen equipo electrónico marítimo periférico. La norma NMEA 0183 define una interface eléctrica y un protocolo de datos para la comunicación entre instrumentos marinos.

Los instrumentos en la norma NMEA 0183 son designados como *parlante* (talkers) o como *oyentes* (listeners), utilizando una interfaz serial asíncrona con los parámetros que se muestran en la tabla 3.1. El NMEA 0183 es un estándar industrial internacional de utilización voluntaria, su primera publicación fue en marzo de 1998 y esta disponible en las oficinas de la NMEA [NMEA 2000].

Baudaje	4800
Número de bits de datos	8
Bits de parada	1
Paridad	Ninguna
Saludo	Ninguno

Tabla 3.1 Parámetros de la interfaz Asíncrona NMEA 0183

3.1 Interfaz Electrónica

La Norma NMEA 0183 permite un *parlante* y varios *oyentes* en un circuito. El cableado de interconexión recomendable es un par trenzado blindado, con el blindaje aterrizado solo en el *parlante*. El estándar no especifica el uso de un conector en particular.

Es recomendable que la salida del *parlante* cumpla con la norma EIA RS-422, que indica dos líneas de señal, "A" y "B". Las señales de control diferencial no tienen referencia a tierra y son prácticamente inmunes al ruido. Esta norma también permite la utilización de una sola línea con niveles TTL, (cuando la salida A es de +5V, la salida B esta en 0V y viceversa). 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 practica el cable sencillo, o el cable A del RS-422 puede ser conectado directamente a la entrada RS-232 de una computadora. De hecho, la mayoría de productos mas recientes, como los receptores GPS de mano, no tienen una salida diferencial RS-422, solo tienen una línea sencilla con niveles TTL o CMOS compatible.

3.2 Formato General de los Mensajes

En la norma NMEA 0183 todos los datos son transmitidos como mensajes. Únicamente se permiten caracteres ASCII despleables en la pantalla. Cada mensaje comienza con el signo "\$" y termina con <CR><LF> (retorno de carro y línea nueva. Existen tres tipos

básicos de mensajes: mensaje de parlante (*talker sentences*), mensajes de propietario (*proprietary sentences*) y mensaje de petición (*query sentences*). El formato general de los mensajes del parlante es:

```
$ttsss,d1,d2,... <CR><LF>
```

Los primeros dos caracteres que siguen al carácter "\$" forman el identificador de parlante, (*talker identifier*). Los siguientes tres caracteres, representados por "sss" en el ejemplo anterior, forman el identificador de mensaje (*sentence identifier*), y son seguidos por un número variable de campos de datos separados por comas, terminando con un verificador de errores (*checksum*) de manera opcional, y finalmente el mensaje termina con <CR><LF>. Los campos de datos son definidos de manera única para cada tipo de mensaje. Un ejemplo de mensaje de parlante es:

```
$HCHDM,238,M<CR><LF>
```

Donde *HC* define al parlante como una brújula magnética; *HDM* especifica el encabezado del mensaje magnético. Los caracteres 238 son el valor del encabezado, y *M* define el valor del encabezado como magnético siendo estos últimos 2 los datos transmitidos.

Un mensaje puede consistir de más de 80 caracteres, aparte de "\$", <CR> y <LF>. Si los datos de algunos de los campos no están disponibles, ese campo será omitido, pero las comas que lo limitan serán transmitidas sin ningún espacio entre ellas. El campo del *checksum* consiste en un "*" y dos dígitos

hexadecimales que representan la operación lógica X-OR sobre todos los caracteres entre "\$" y "*" sin incluir a estos 2 últimos

3.3 Formato del Mensaje NMEA Utilizado

El identificador de parlante o *talker* que se utiliza para el Sistema de Posicionamiento Global es *GP*. Los receptores GPS son capaces de transmitir varios tipos de mensajes, cada uno de los cuales contienen diferente información. Como ejemplos tenemos el mensaje *VTG*, que reporta el curso y la velocidad en la tierra del receptor; el mensaje *GLL* que contiene la posición geográfica (latitud y longitud). Este proyecto utiliza únicamente el mensaje *GGA* ya que es el único que proporciona posición y altitud a un mismo tiempo. El formato del mensaje es el siguiente:

GGA especifica la trama con datos de tiempo, posición y datos de calidad de las lecturas GPS.

```
$GPGGA,hhmmss,xxxx.xxx,a,yyyyy.yyy,b,q,ss,oo.o,hhhh,M,,M,ddd,dddd*cc  
      1      2      3  4      5 6 7  8      9  10  11  12  13
```

Los campos se definen como sigue:

1. Tiempo UTC
2. Latitud
3. N o S (Norte o Sur)
4. Longitud
5. E o W (Este u Oeste)
6. Indicador de calidad GPS (0 = datos no disponibles, 1 = GPS fijo, 2 = DGPS fijo)
7. Numero de satélites a la vista (de 00 a 12)
8. Dilución horizontal de precisión (HDOP)
9. Altitud
10. Unidades de la altitud

- 11. Datos del GPS diferencial
- 12. Identificación de estación de referencia diferencial
- 13. checksum

Los principales Campos de Interés para este proyecto son

Tabla 3.2 datos útiles del mensaje GPGGA en este proyecto

No. De Campo & Información	No. De caracteres	Formato de información
2) latitud	7 xxxx.xxx	ggmm.ddd
3) N o S	1 a	-
4) Longitud	8 yyyyyy.yyy	Gggmm.ddd
5) E o W	1 b	-
9) Altitud	4 hhhh	nxxx

*g = grados, m = minutos, d = décimas de minuto, n= metros

No.	Bit 76543210	Description	Example	
			Setting Value	Output Data (HEX)
1	0010 0100	Header "\$"	Fixed	24
2		Address field	Fixed	
3	0100 0111	"G"		47
4	0101 0000	"P"		50
5	0100 0111	"G"		47
6	0100 0111 0100 0001	"G" "A"		47 41
7	01001100	","	Fixed	2C
		UTC time	06:22:43	
8	xxxxxxx	Hour (10's digit)		00
9	xxxxxxx	Hour (1's digit)		36
10	xxxxxxx	Minute (10's digit)		32
11	xxxxxxx	Minute (1's digit)		32
12	xxxxxxx	Second (10's digit)		34
13	xxxxxxx	Second (1's digit)		33
14	01001100	","	Fixed	2C
		Latitude	36° 03.979'	
15	xxxx xxxx	Degree (10's digit)		33
16	xxxx xxxx	Degree (1's digit)		36
17	xxxx xxxx	Minute (10's digit)		30
18	xxxx xxxx	Minute (1's digit)		33
19	0010 1110	","		2E
20	xxxx xxxx	Minute (0.1's digit)		39
21	xxxx xxxx	Minute (0.01's digit)		37
22	xxxx xxxx	Minute (0.001's digit)		39
23	0100 1100	","	Fixed	2C
24	xxxx xxxx	Latitude direction "N" or "S"	North latitude	4E
25	0100 1100	","	Fixed	2C
		Longitude	140° 10.296'	
26	xxxx xxxx	Degree (100's digit)		31
27	xxxx xxxx	Degree (10's digit)		34
28	xxxx xxxx	Degree (1's digit)		30
29	xxxx xxxx	Minute (10's digit)		31
30	xxxx xxxx	Minute (1's digit)		30
31	0010 1110	","		2E
32	xxxx xxxx	Minute (0.1's digit)		32
33	xxxx xxxx	Minute (0.01's digit)		39
34	xxxx xxxx	Minute (0.001's digit)		36

Tabla 3.3.1 Campos de información en el mensaje GPGGA

No.	Bit 76543210	Description	Example	
			Setting Value	Output Data (HEX)
35	0100 1100	","	Fixed	2C
36	xxxx xxxx	Longitude direction "E" or "W"	West longitude	57
37	0100 1100	","	Fixed	2C
38	xxxx xxxx	GPS Quality Indicator "0": Invalid "1": GPS measurement "2": D-GPS measurement	D-GPS measurement	32
39	0100 1100	","	Fixed	2C
40	xxxx xxxx	No. of satellites used for measurement calc. "00" to "12"	7-satellite measurement	30
41	xxxx xxxx			37
42	0100 1100	","	Fixed	2C
		HDOP	1.2	
43	xxxx xxxx	10's digit		30
44	xxxx xxxx	1's digit		31
45	0010 1110	."		2E
46	xxxx xxxx	0.1's digit		32
47	0100 1100	","	Fixed	2C
		Altitude (m)	23m	
48	xxxx xxxx	1,000's digit		30
49	xxxx xxxx	100's digit		30
50	xxxx xxxx	10's digit		32
51	xxxx xxxx	1's digit		33
52	0100 1100	","	Fixed	2C
		Altitude units	Fixed	
53	0100 1101	"M"		4D
54	0100 1100	","	Fixed	2C
55	0100 1100	","	Fixed	2C
56	0100 1101	"M"	Fixed	4D
57	0100 1100	","	Fixed	2C
		D-GPS data elapsed time (s)	5s	
58	xxxx xxxx	100's digit		30
59	xxxx xxxx	10's digit		30
60	xxxx xxxx	1's digit		35
61	0100 1100	","	Fixed	2C
		D-GPS reference station ID	0	
62	xxxx xxxx	1,000's digit		30
63	xxxx xxxx	100's digit		30
64	xxxx xxxx	10's digit		30
65	xxxx xxxx	1's digit		30
66	0010 1010	"*"	Fixed	2A

Tabla 3.3.2 Campos de información en el mensaje GPGGA cont.

No.	Bit 76543210	Description	Example	
			Setting Value	Output Data (HEX)
67	xxxx xxxx	Checksum Hexadecimal upper digits	4A	34
68	xxxx xxxx	Hexadecimal lower digits		41
69	00001101	Terminator <CR>	Fixed	0D
70	00001010	<LF>		0A

Tabla 3.3.3 Campos de información en el mensaje GPGGA cont

3.4 Referencias

National Marine Electronic Association; "NMEA 0183 Protocol";
Versión 2.30; Enero 2000 <http://www.nmea.org>

Valence Semiconductor Inc.
<http://www.valencesemi.com/>

4. VEHÍCULO Y SISTEMA DE CONTROL	1
4.1 SISTEMA DE CONTROL	1
4.1.1 Receptor GPS.....	1
4.1.2 Microcontrolador MC68HC11F1.....	5
4.1.2.1 Definición de Microcontrolador.....	6
4.1.2.2 Especificaciones Técnicas del Microcontrolador MC68HC11F1.....	7
4.1.2.3 Puerto de Comunicación Serial.....	7
4.1.2.4 Registros del Microcontrolador MC68HC11F1.....	9
4.1.2.5 Puertos del Microcontrolador MC68HC11F1.....	12
4.1.2.6 Memoria del Microcontrolador MC68HC11F1.....	13
4.1.3 Driver o Controlador.....	15
Funcionamiento del Optoacoplador.....	17
4.2 VEHÍCULO.....	18
4.2.1 Vehículo.....	18
4.3 REFERENCIAS.....	20

4. Vehículo y Sistema de Control

En este capítulo se describe el sistema de control propuesto para este proyecto, tanto en lo que respecta a su unidad de control como a la interfaz con lo cual se conecta al vehículo.

4.1 Sistema de control

El sistema de control está constituido por 3 partes esenciales: un Receptor GPS (sistema de posicionamiento) el cual provee la posición exacta del vehículo en todo momento, un microcontrolador que se encarga de realizar los cálculos de dirección y trayectoria a partir de la posición y así generar las señales de control de movimiento, y un driver o controlador que a partir de la señal lógica producida por el microcontrolador acciona los motores que mueven el vehículo

4.1.1 Receptor GPS.

En el desarrollo de este proyecto se utilizó un receptor GPS Modelo VM7002 fabricado por la compañía *Valence Semiconductors* en Silicon Valley California EU [Valence 2002]. Este receptor cuenta con dos tipos de circuitos integrados GPS, uno manufacturado por la empresa Sony y el otro por Valence (VS7001) cada uno utiliza un formato diferente. El circuito integrado Sony utiliza un formato propietario llamado Sony New y mientras que el circuito integrado Valence utiliza el formato del estándar NMEA 0183. En este proyecto se utiliza exclusivamente el circuito integrado

valence (VS7001) ya que utiliza el formato del estándar voluntario NMEA 0183 descrito en el capítulo anterior.

El modulo VM7002 es un receptor de alto desempeño de 16 canales, tiene un tamaño compacto (figura 4.1) y un consumo muy bajo de potencia que opera con voltajes entre 3.3 V y 3.6 V con un rango de temperatura de operación entre -40° C. y $+85^{\circ}$ C. Además cuenta con una batería recargable de ion-litio que permite una recepción de la señal de manera casi ininterrumpida, una interfase serial, conexión para antena activa, y soporta GPS Diferencial DGPS, estas son algunas de las características que presenta el receptor GPS utilizado

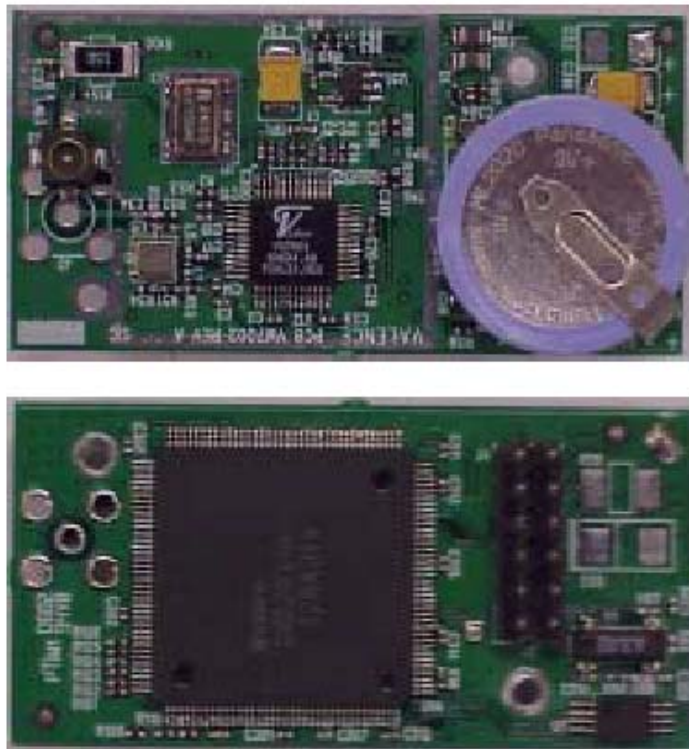


figura 4.1 Modulo VM7002

Este receptor permite gracias a sus 16 canales paralelos la recepción de 12 satélites al mismo tiempo, lo cual junto con la corrección DGPS permite el calculo de una posición con una gran exactitud. En condiciones climatológicas adversas permite el calculo de la posición con la recepción únicamente de tres satélites y aunque no es muy exacta es muy útil en este tipo de situaciones. Además, cuenta con una gran rapidez para readquirir la señal cuando esta se ha perdido, es decir, cuando se atraviesa un puente o una zona con un grueso follaje.

El receptor utiliza el sistema de posicionamiento global GPS de la constelación NAVSTAR con señal de portadora L_1 a una frecuencia de 1.57542 GHz. y con el condigo C/A

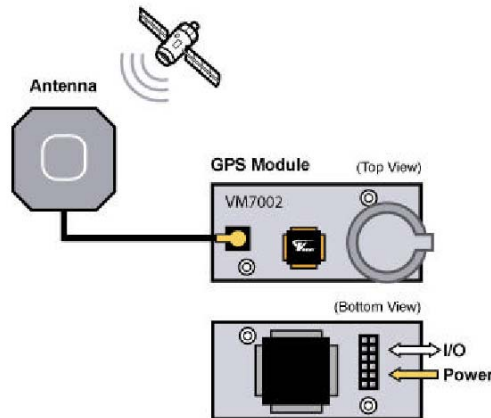


figura 4.2 Diagrama del receptor GPS

A continuación se presentan tablas que muestran las especificaciones técnicas y la asignación de pines del receptor GPS modulo VM7002 fabricado por Valence Semiconductors Inc.

Tabla 4.1 especificaciones técnicas del modulo VM7002

Rastreo de satélite	16 canales paralelos	
RF de entrada y código	Frecuencia Central	1.57542 GHz en la banda L1

	Impedancia	50 ohm
	Sensibilidad de la señal	-138 dBm o menos
Sistema de Posicionamiento	Default	WGS-84
Exactitud de posicionamiento	Posición	30 m 2DRMS PDOP=2.5 HDOP=1.5
	Velocidad	0.3 m/s PDOP=2.5 HDOP=1.5
Condición de posicionamiento		a) DOP limite: 3D:PDOP < 12 2D:HDOP < 6 b) Elevación: 5 ^a
Limite de desempeño	Velocidad	500 Km/h o menos
	Aceleración	2G o menos
tiempo de actualización de datos		1 segundo
Función DGPS		DARC BTA R-003 standard RTCM SC1 04 version 2.1 (formato de 6 a 8) Tipo 1 dato para corrección de calculo
Tipo de navegación		12 Satélites visibles 2 Satélites visibles
Tasa de Adquisición de datos	Encendido Caliente	6 a 20 segundos promedio
	Encendido Tibio	33 a 50 segundos promedio
	Encendido Frío	35 a 55 segundos promedio
Readquisición de datos	perdida < 5 minutos	3 a 6 segundos
	perdida > 5 minutos	6 a 16 segundos

Tabla 4.3 especificaciones de comunicación

Método de comunicación	Serial
Tasa de transmisión entrada/salida	4800bps NMEA 0183
	9600bps Sony New
Niveles lógicos	TTL
Código I/O	ASCII
Formato de comunicación	NMEA 0183 o Sony New

Tabla 4.2 Asignación de pines del receptor VM7001

**I/O Connector Pins- J5
Pin Configuration**

Pin No.	Symbol	I/O	Description
1	Vdd	-	Digital VDD and battery charge power input.
2	Vdd	-	Main power input. 5-12V input if regulator option is used
3	GND	-	Ground
4	Reset#	I	Module reset input.
5	RXD1	I	Differential corrections input.
6	NC	-	No connection
7	GND	-	Ground
8	TXD0	O	Navigation data output.
9	RXD0	I	Command Input
10	GND	-	Ground

**Mode Control Connector- J4
Pin Configuration**

Pin No.	Symbol	I/O	Description
1	MODE	I	Communications format. (L = Sony, H = NMEA 0183)
2	GND	-	Ground

4.1.2 Microcontrolador MC68HC11F1.

A continuación se describe el microcontrolador utilizado en este proyecto.

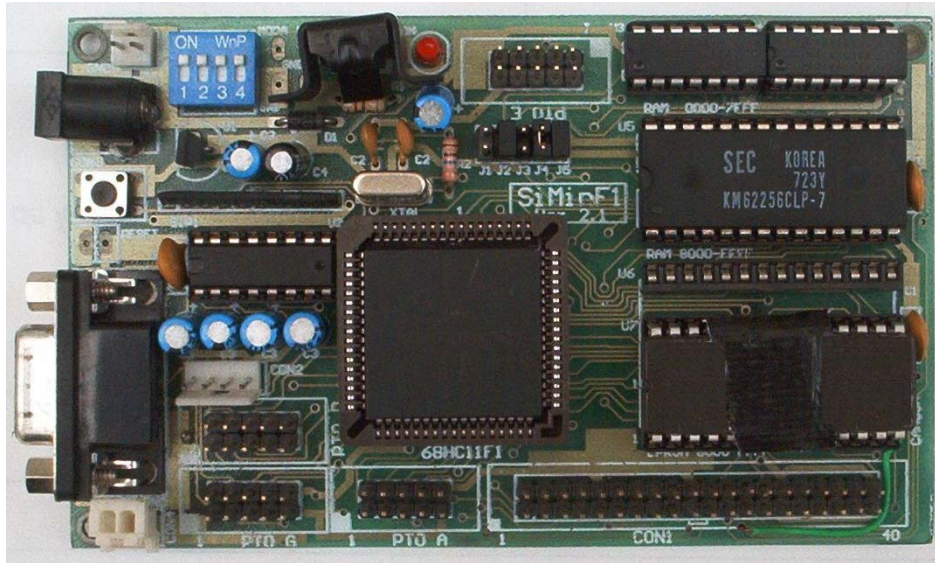


Figura 4.3 Microcontrolador MC68HC11F1

4.1.2.1 Definición de Microcontrolador

Un microcontrolador (MCU) es un circuito integrado que incorpora una unidad central de proceso (CPU) con una serie de recursos internos. El CPU permite que el microcontrolador pueda ejecutar instrucciones almacenadas en una memoria. Los recursos internos son memoria RAM, memoria ROM, memoria EEPROM, puerto serie, puertos de entrada/salida, temporizadores, comparadores, capturadores, etc. El microcontrolador es una evolución de los microprocesadores, ya que se integran al microprocesador las funciones que antes era necesario situar externamente con otros circuitos. Un ejemplo típico de esta evolución son los puertos de entrada/salida y la memoria RAM, puesto que en los sistemas con microprocesadores era necesario desarrollar una lógica de control y unos circuitos para implementar las funciones anteriores, en un microcontrolador no es necesario ya que lo tiene todo integrado

En resumen, un microcontrolador es un circuito integrado independiente, que no necesita memoria ni puertos externos pues

los lleva en su interior, que facilita la tarea de diseño y reduce el espacio, traduciéndose todo a una aplicación final más económica y fiable.

4.1.2.2 Especificaciones Técnicas del Microcontrolador MC68HC11F1

El microcontrolador MC68HC11F1 cuenta con una unidad central de proceso (CPU) M68HC11 fabricado por motorola, con las siguientes especificaciones:

- o 512 bytes de memoria EEPROM
- o 1024 bytes de memoria RAM
- o 7 puertos paralelos (A-G)
- o 96 registros programables
- o 8 canales de 8 bits de resolución
- o Bus de datos con direcciones no multiplexadas
- o Temporizador de 16- bit
- o 3 canales de funciones de captura de entrada (ICX)
- o 4 canales de funciones de comparación de salida (OCX)
- o 1 canal extra que puede utilizarse como comparador de salida
 - o captura de entrada
- o Acumulador de pulsos de 8 bits
- o Interrupción en tiempo real (RTI)
- o Puerto de comunicación serial (SCI)
- o Puerto de comunicación con periféricos (SPI)
- o 8 canales de convertidor analógico digital A/D.

4.1.2.3 Puerto de Comunicación Serial

El microcontrolador 68HC11 dispone de una unidad de comunicaciones serie (SCI) que permite realizar comunicaciones asíncronas a distintas velocidades y con paquetes de 8 y 9 bits. (8 ó 9 bits de datos). Los paquetes mandados contienen un bit al

principio del encabezado (Bit de start), 8 ó 9 bits de datos y un bit de parada (bit stop) al final.

Para configurar los parámetros de comunicación, envío y recepción datos, y comprobar el estado de la transmisión, el SCI dispone de 5 registros mapeados en memoria. Además, el SCI soporta el modo especial de funcionamiento WAKE-UP, usado en sistemas multireceptores poco frecuente.

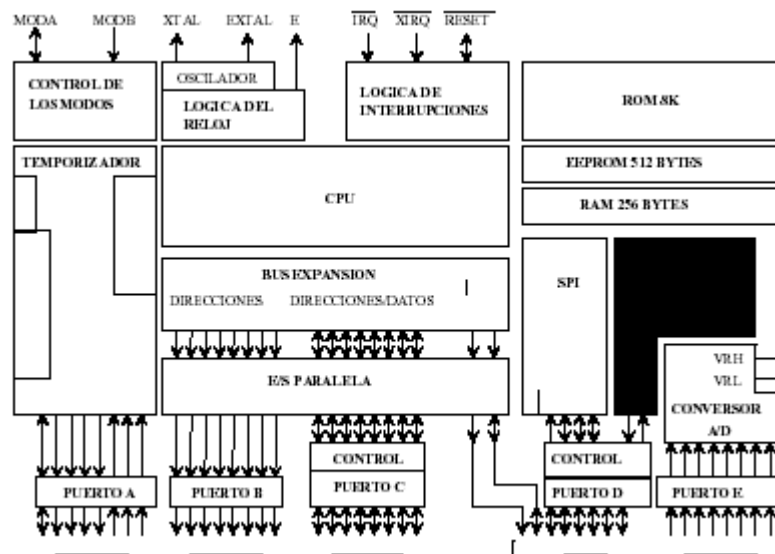


Figura 4.4 diagrama de bloques de microcontrolador MC68HC11F1

El SCI está formado por una unidad de transmisión y una unidad de recepción que son totalmente independientes, lo que permite que las comunicaciones sean bidireccionales. La unidad de transmisión está formada por un registro de desplazamiento con carga en paralelo, llamado registro de transmisión. Al introducir un valor en este registro, comienza a desplazarse hacia la derecha el contenido del registro, enviando los bits por la línea serie, a una velocidad configurable por el usuario. De la misma manera la unidad de recepción dispone de otro registro, registro de recepción que recibe los bits en serie y los va desplazando

hasta obtener un dato en paralelo que puede ser leído. Tanto el registro de transmisión como el de recepción están mapeados en la misma dirección de memoria. Al escribir en esa dirección de memoria, el dato se cargará en el registro de transmisión. Al efectuar una lectura, el dato se leerá del registro de recepción. Ambos registros comparten la misma dirección física de memoria pero se trata de dos registros diferentes. Puesto que ambos registros comparten dirección física, se les ha asignado un único nombre: registro de datos (SCDR), y está situado en la dirección \$102F.

Una vez configurado el SCI adecuadamente, enviar y recibir datos es una tarea muy sencilla: basta con leer o escribir en el registro de datos.

4.1.2.4 Registros del Microcontrolador MC68HC11F1

El CPU del microcontrolador dispone de 2 registros A y B acumuladores de 8 bits, que se unen para formar el registro D de 16 bits, siendo el acumulador A la parte alta y el acumulador B la parte baja. Además dispone de 2 registros para direccionamiento indexado X,Y ambos de 16 bits. El puntero de pila y el contador de programa son también de 16 bits, lo que permite que la longitud máxima de un programa sea de 64Kbytes, que es el espacio máximo direccionable por el microcontrolador. El registro CCR es el llamado registro de estado, que contiene unos bits que reflejan el estado del CPU



Figura 4.5 registros de microcontrolador MC68HC11F1

El **puntero de pila** de memoria debe ser inicializado por el usuario. La pila crece desde direcciones altas hacia direcciones bajas, por lo que al introducir un elemento en la pila, el SP se decrementa en 1 ó 2 bytes dependiendo del tamaño del dato ingresado en la pila. Al sacar un elemento de la pila, SP se incrementa.

El contador de programa **PC** se va incrementado según se van ejecutando las instrucciones. Por tanto, los programas se ejecutan desde direcciones bajas a altas y la pila crece desde direcciones altas a bajas. Es importante dar a SP un valor seguro de tal manera que la pila no se traslape con el código, si es que el código se encuentra en RAM.

El registro **CCR** es de 8 bits. Cada bit tiene una letra asignada y representa una situación diferente del estado del CPU.

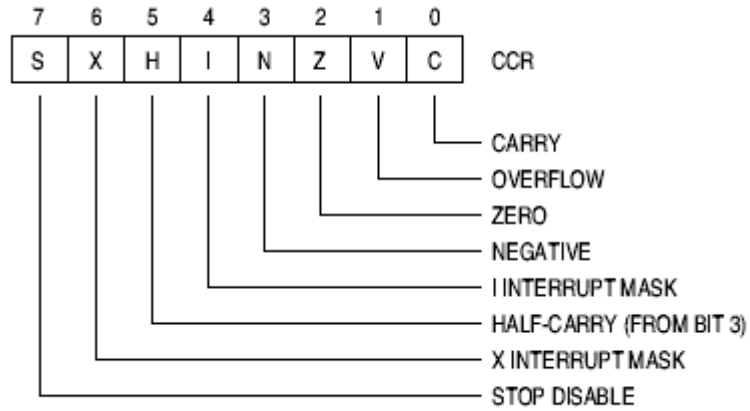


Figura 4.6 Registro CCR

El registro de código de condición (CCR) contiene cinco indicadores (C,V,Z,N,H), dos bits de enmascaramiento de interrupción (IRQ y XIRQ) y un bit de deshabilitación de paro o stop (S).

Carry (acarreo) (C).- Este bit se enciende si la unidad lógica aritmética (ALU) realiza un acarreo

Overflow (sobreflujo) (V).- Este bit se enciende si alguna operación causa sobre flujo aritmético.

Zero (cero) (Z).- Este bit se enciende si el resultado de alguna operación, ya sea aritmética, lógica o alguna manipulación de datos, es cero.

Negative (negativo) (N).- Este bit se enciende si el resultado de alguna operación, ya sea aritmética, lógica o alguna manipulación de datos, es negativo, o sea que el bit más significativo sea uno.

Half Carry (medio acarreo) (H).- Este bit se enciende cuando ocurre un acarreo entre los bits 3 y 4 de la unidad lógica aritmética (ALU).

Stop Disable (deshabilitado de Alto) (S).- Al encender este bit se previene la ejecución de la instrucción STOP, la cual coloca al microcontrolador en modo de alto con baja potencia donde todos los relojes y el oscilador se detienen

Interrupt Mask (Mascara de interrupción) (I) La mascara (bit I) de petición de interrupción deshabilita las fuentes de interrupción.

X Interrupt Mask (Mascara de interrupción X) (X)este bit deshabilita las interrupción que provienen del pin XIRQ

4.1.2.5 Puertos del Microcontrolador MC68HC11F1

Puerto A: de 8 bits mapeado en la dirección \$1000 cuenta con 8 canales bidireccionales de propósito general que pueden ser utilizados indistintamente como entrada o salida programables en el registro DDRA mapeado en la dirección \$1001. Comparte funciones con los comparadores de salida, los capturadores de entrada y el temporizador de Tiempo.

Puerto B: de 8 bits mapeado en la dirección \$1004 cuenta con 8 canales unidireccionales de salida de propósito general en el modo expandido funciona como la parte alta del bus de direcciones de memoria.

Puerto C: de 8 bits mapeado en la dirección \$1006 cuenta con 8 canales bidireccionales de propósito general que pueden ser utilizados indistintamente como entrada o salida programables en el registro DDRC mapeado en la dirección \$1007. En el modo expandido funciona como el bus de datos de la memoria.

Puerto D: de 6 bits mapeado en la dirección \$1008 cuenta con 8 canales bidireccionales de propósito general que pueden ser utilizados indistintamente como entrada o salida programables en el registro DDRD mapeado en la dirección \$1009. Comparte funciones con el Puerto de comunicaciones Serial (SCI) y con el puerto de comunicación con periféricos (SPI)

Puerto E: de 8 bits mapeado en la dirección \$100A cuenta con 8 canales unidireccionales de entrada de propósito general. Comparte funciones con el convertidor analógico digital A/D

Puerto F: de 8 bits mapeado en la dirección \$1005 cuenta con 8 canales unidireccionales de salida de propósito general en el modo expandido funciona como la parte baja del bus de direcciones de memoria.

Puerto G: de 8 bits mapeado en la dirección \$1002 cuenta con 8 canales bidireccionales de propósito general que pueden ser utilizados indistintamente como entrada o salida programables en el registro DDRG mapeado en la dirección \$1003.

4.1.2.6 Memoria del Microcontrolador MC68HC11F1

El MCU direcciona 64Kbytes de memoria, parte de esta memoria se encuentra dentro del MCU y el resto se puede implementar mediante chips de memoria externos al MCU. La memoria RAM se

sitúa por defecto a partir de la dirección \$0000 hasta la \$03FF (1K bytes de RAM). De la dirección \$1000 hasta la \$103F se encuentran situados los registros de control del MCU. Estos registros son células de memoria RAM o EEPROM (no volátiles). Las células EEPROM sólo se pueden escribir bajo circunstancias especiales, y siempre que el MCU esté en modo especial. Entre las direcciones \$FE00-\$FFFF se sitúan 512 bytes de memoria EEPROM que están disponibles en el modelo F1. En los modos especiales se mapea una memoria ROM en las direcciones \$BF40-\$BFFF que contiene el programa BOOTSTRAP usado para cargar programas externos y una tabla con los vectores de interrupción. Desde la dirección \$FFC0 hasta la \$FFFF se encuentra la tabla de vectores de interrupción en el modo normal.

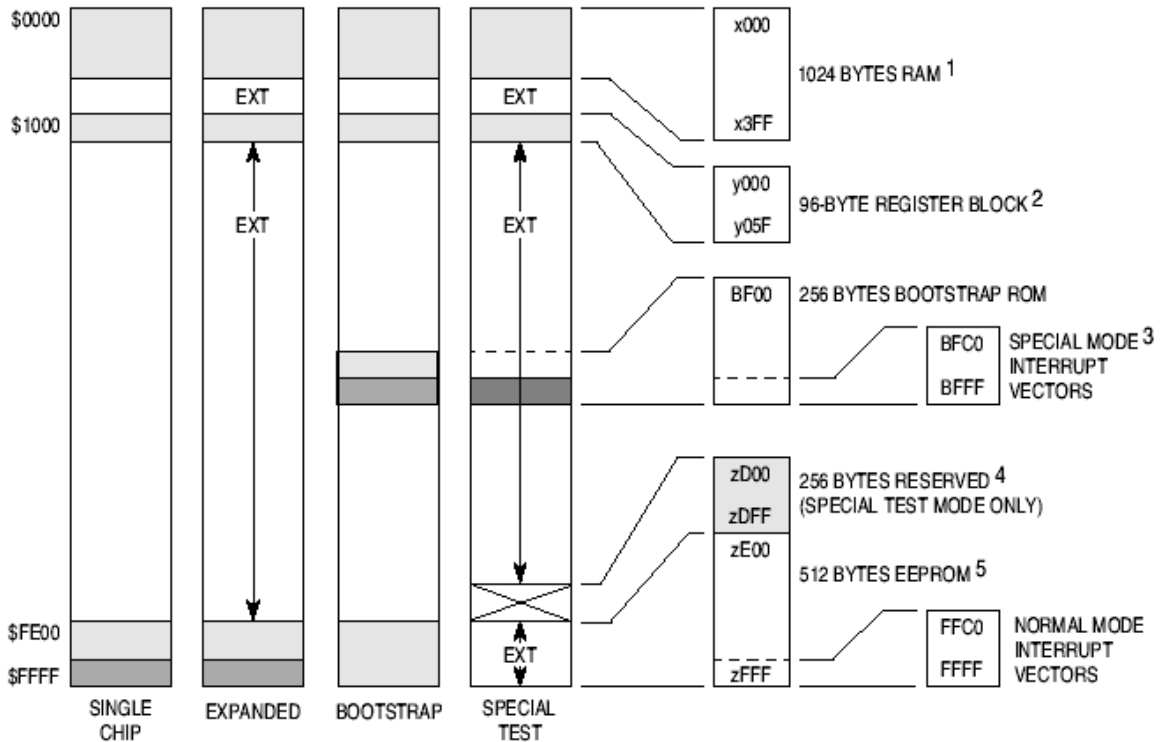


Figura 4.7 mapa de memoria del microcontrolador MC68HC11F1

Configurando algunos registros de control adecuadamente se puede remapear el sistema para ajustarlo a las necesidades de la aplicación. Por ejemplo, la memoria RAM se puede situar en cualquier otra parte (con alguna restricción) dentro de los 64Kbytes del mapa de memoria. Lo mismo ocurre con los registros de control.

El registro que permite cambiar las direcciones de la memoria RAM y de los registros de control se denomina INIT y se encuentra en la dirección \$103D.

Este microcontrolador puede funcionar con 4 distintos modos de operación dos normales y 2 especiales, los normales son. *Single chip* y *modo expandido* y los especiales son el *boot strap* y el *special test*; para cada uno de estos modos de operación la memoria se reconfigura de acuerdo a la figura 4.7

4.1.3 Driver o Controlador

Para el control de motores se utilizo un circuito que hace las funciones de driver o controlador, esto es a partir de las señales digitales emitidas por el microcontrolador este controlador enciende o apaga los motores del vehículo según se requiera el caso.

En este caso se diseño un circuito el cual puede controlar 3 grupos de motores diferentes, para esto se utilizan 2 circuitos L293 o drivers el cual puede controlar motores de 1V a 36V con señales TTL, y 5 optoacopladores para aislar la fuente de alimentación de los motores del microcontrolador,

Driver: el L293 es un driver de 4 canales el cual proporciona una salida de 1A a 2A, inmune al ruido y con protección al sobrecalentamiento. Cada uno de estos circuitos es capaz de controlar 4 motores de corriente directa, uno por cada canal, cuando los motores solo se mueven en una dirección (en encendido y apagado), o controlar 2 motores utilizando dos canales por motor pero pudiendo invertir la dirección del giro del motor.

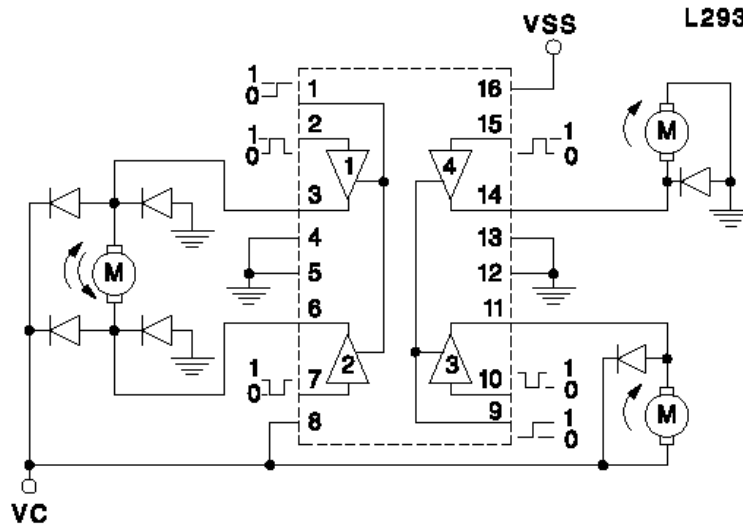


Figura 4.8 patigrama del circuito L293

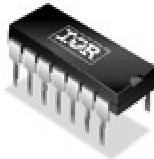


Figura 4.9 encapsulado tipo DIP del L293

Debido a que la fuente de alimentación de los motores es variable entre 1V y 36V se necesita aislar el microcontrolador del driver, para lo que se utiliza un optoacoplador el cual recibe una señal TTL y ópticamente habilita un bit de otra fuente que puede ser de otro valor de voltaje lo que aísla una de la otra

Un optoacoplador combina un dispositivo semiconductor formado por un fotoemisor, un fotoreceptor y entre ambos hay un camino por donde se transmite la luz. Todos estos elementos se encuentran dentro de un encapsulado que por lo general es del tipo DIP.



Figura 4.10 encapsulado tipo dip

Funcionamiento del Optoacoplador

La señal de entrada es aplicada al fotoemisor y la salida es tomada del fotoreceptor. Los optoacopladores son capaces de convertir una señal eléctrica en una señal luminosa modulada y volver a convertirla en una señal eléctrica. La gran ventaja de un optoacoplador reside en el aislamiento eléctrico que puede establecerse entre los circuitos de entrada y salida.

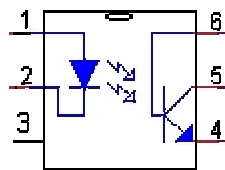


Figura 4.11 patigrama de un optoacoplador tipo dip

Los fotoemisores que se emplean en los optoacopladores de potencia son diodos que emiten rayos infrarrojos (IRED) y los fotoreceptores transistores.

Cuando aparece una tensión sobre los terminales del diodo IRED, este emite un haz de rayos infrarrojo que transmite a

través de una pequeña guía-ondas de plástico o cristal hacia el fotorreceptor. La energía luminosa que incide sobre el fotorreceptor hace que este genere una tensión eléctrica a su salida. Este responde a las señales de entrada, que podrían ser pulsos de tensión.

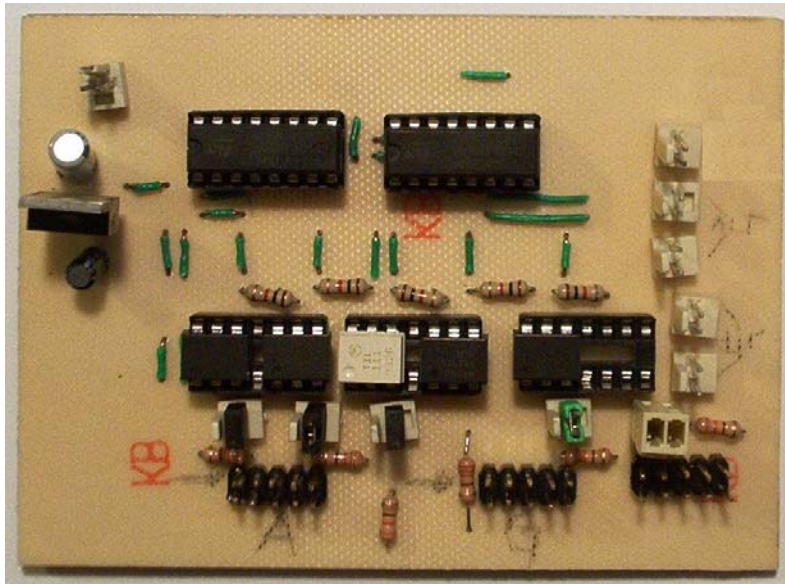


Figura 4.12 Circuito Driver Diseñado

4.2 Vehículo

El objetivo primordial de este proyecto es el sistema de control de navegación, es por ello que, el tipo de vehículo pasa a segundo plano. Sin embargo, el diseño del sistema de control, se realizó, de tal forma que puede ser adaptable a cualquier tipo de vehículo que se desplace sobre la superficie terrestre o por el aire.

4.2.1 Vehículo.

El sistema de control de navegación esta diseñado para realizar correcciones de trayectoria en 3 dimensiones, por lo

tanto, tratándose de un vehículo que se desplace sobre la superficie terrestre, el sistema puede ser utilizado de igual manera, únicamente despreciando las señales de control de altitud.

Tratándose de un vehículo que se desplace sobre la superficie terrestre, el sistema funciona perfectamente puesto que, tratándose de un auto o un barco, las velocidades promedio de los mismos no requieren velocidades de procesamiento muy altas por parte del sistema.

En el caso, de vehículos aéreos se tienen 2 tipos: los que se desplazan a velocidades muy altas como lo son misiles y aviones, y los aerostatos o dirigibles de gas cuyas velocidades son comparables con las de los vehículos terrestres. El sistema esta diseñado para vehículos aéreos de velocidades pequeñas debido a la velocidad de proceso del microcontrolador utilizado.

4.3 Referencias

Valence Semiconductor Inc.

<http://www.valencesemi.com/>

Motorola, Inc.

<http://www.motorola.com/>

5.	ALGORITMO DE NAVEGACIÓN.	1
5.1	POSICIONAMIENTO	1
5.2	PLANIFICACIÓN DE TRAYECTORIAS	4
5.3	ALGORITMO DE NAVEGACIÓN DESARROLLADO	7
5.3.1	ESTRUCTURA DEL ALGORITMO DE NAVEGACIÓN	9
5.4	REFERENCIAS	32

5. Algoritmo de Navegación.

Se considera el termino "Navegación" como el desplazamiento controlado de un móvil a través de un espacio definido. La navegación automática existe cuando el sistema de navegación del móvil no recibe señales de control externas si no que este produce sus propias señales de control internamente. El tipo de sistema de navegación va a depender de las características del espacio en el cual el móvil se desplazará. Para establecer la navegación hay cuatro elementos básicos necesarios presentes en todos los sistemas de navegación autónoma.

- Marco de referencia de posicionamiento
- Mecanismo para posicionamiento
- Unidad de control (procesamiento)
- Subsistemas para desplazamiento

Tomando como ejemplo cuando un robot debe desplazarse en el área de trabajo de una fabrica, se pueden utilizar marcas luminosas, metálicas o magnéticas en el piso las cuales el robot debe ser capaz de detectar o reconocer para ubicarse en el espacio en el que se desplaza, dichas marcas constituyen el marco de referencia de posicionamiento del robot, los mecanismos mediante los cuales el robot reconoce estas marcas o marco de referencia forman el mecanismo para el posicionamiento

5.1 Posicionamiento

El sistema de referencia WGS-84ⁱ fue desarrollado y difundido por la agencia norteamericana *National Imagery and Mapping Agency (NIMA)* y adoptado por la ICAOⁱⁱ como estándar internacional en 1989.[WGS-84 1997] Este sistema en la actualidad prácticamente esta basado en las levantamientos GPS. La versión más reciente del sistema WGS-84 es la denominada versión G873, de 1997 donde la letra "G" indica que la versión solo contiene levantamientos GPS. El número 873 indica el sistema GPS en el cual las efemérides precisas calculadas por la NIMA fueron distribuidas por primera vez al publico en este nuevo sistema coordinado (0h UTC, 29 septiembre de 1996). Las efemérides incluidas en el mensaje radiado por los satélites GPS se expresan también en este marco de referencia desde el 29 de enero de 1997. Hasta entonces se había utilizado el sistema WGS84(G730), definido de forma similar.

El punto de origen de este sistema se encuentra cercano a la intersección entre el meridiano de Greenwich y el eje Ecuatorial, lo cual da origen a los cuatro cuadrantes que son utilizados para el posicionamiento en cualquier punto sobre la superficie terrestre.

ⁱ World Geodetic System 1984

ⁱⁱ International Civil Aviation Organisation

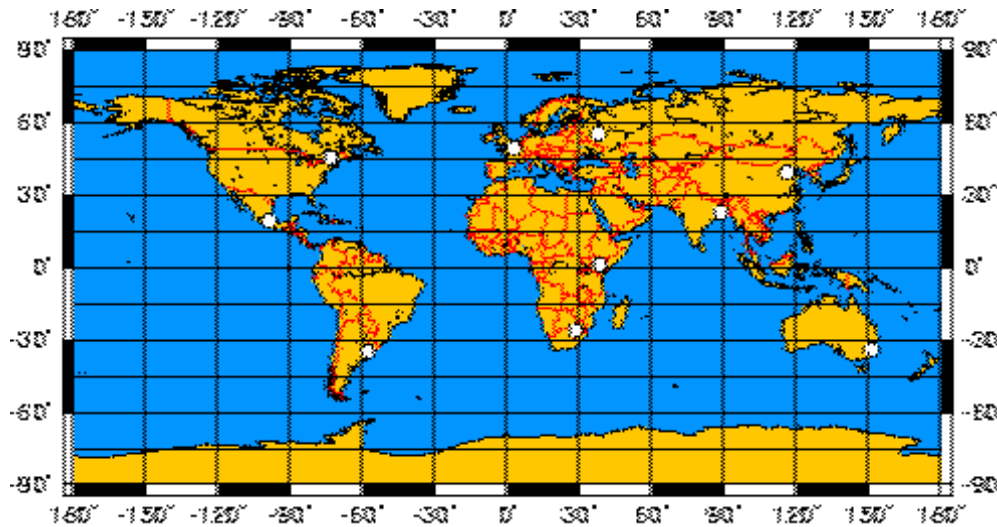


Figura 5.1 Mapa Global Referido al sistema WGS-84

En el sistema de referencia WGS-84 cada punto de la superficie terrestre posee una posición definida a partir de un par de valores conocidos como coordenadas terrestres: latitud y longitud, mas un carácter asociado a cada valor Norte o Sur y Este u Oeste respectivamente.

Ya que en este proyecto se planeo para un vehículo pequeño de baja velocidad el cual no estaría equipado para el desplazamiento a través de grandes distancias, se diseñó de manera que pueda desplazarse en el cuadrante en el cual nos ubicamos NW o Noroeste del mapa ya que de ésta forma se ahorran recursos de procesamiento. De esta manera, se hace hincapié en que tanto el algoritmo de navegación como el programa pueden ser modificados ligeramente para poder utilizar el vehículo en otro cuadrante del mapa.

Tomando en cuenta que el perímetro de la tierra es aproximadamente de 40,350 Km, un grado de latitud o de longitud equivale a 113 Km y un minuto a 1.8 Km. Además del planteamiento del proyecto para recorrer pequeñas distancias, solo se tomaran

en cuenta únicamente minutos y décimas de minutos. Sin embargo, para un proyecto mas completo y ambicioso planeado para grandes distancias basta con realizar ligeras modificaciones en el algoritmo de navegación y en el programa de control.

En este proyecto se utilizara el cuadrante NW y las diferencias entre este cuadrante y los ejes coordenados utilizados comúnmente son las siguientes: En los ejes coordenados utilizados comúnmente las unidades en el eje X van de negativo a positivo a medida que se avanza de oeste a este y al utilizar el cuadrante NW se tiene que en el eje X los valores menores se encuentran al este y aumentan cuando se avanza de este a oeste por lo que un incremento positivo en la latitud en el marco de referencia de ejes coordenados será negativo ya que se desplazo de oeste a este o sea de mayor a menor.

5.2 Planificación de trayectorias

Una vez que se tiene el sistema de posicionamiento del móvil resuelto, el siguiente paso es la planificación de la trayectoria que el móvil debe seguir para llegar a su destino. Todos los métodos parten del siguiente escenario básico: dado un robot **R** y un entorno que contiene un conjunto de obstáculos fijos, encontrar un camino desde una *configuración* inicial **S** hasta una configuración final **G**, tal que **R** pueda viajar libremente desde **S** hasta **G** sin colisionar con ninguno de los obstáculos. Una configuración consiste en posición-orientación para el robot R en un momento dado.

La planificación de movimientos ha recibido considerable atención durante los últimos años. Los métodos que existen para esta tarea se pueden dividir en tres categorías: métodos de

descomposición en celdas, métodos de mapa de caminos(road-map) y métodos de campo potencial[Guardiola, 2000] Estos métodos pueden actuar bien en el *espacio de trabajo* del robot, o sea, el espacio en el que están definidos el robot y los obstáculos, o en el *espacio de configuración*, el espacio que consiste en todas las posibles configuraciones del robot. Una ventaja de este último es que cualquier configuración para el robot consiste simplemente en un sólo punto en este espacio y así el problema de planificar el movimiento del robot entero se reduce a planificarlo para un punto.

Descomposición en celdas. El método de descomposición en celdas construye un camino en el espacio de configuración dividiendo el espacio libre , esto es, la parte del espacio de configuración donde el robot no colisiona con ningún obstáculo en simples celdas; esto puede hacerse de forma exacta o aproximada. Después, las celdas son insertadas en un grafico en el que celdas adyacentes están conectadas. Entre estas celdas, el movimiento es normalmente fácil de calcular. El movimiento que conecta la configuración inicial y la final puede entonces calcularse al determinar las celdas que contienen las configuraciones inicial y final, encontrando un camino entre estas celdas en el grafico, y finalmente calculando el movimiento entre cada arista del camino en el grafico.

Mapa de caminos. El método de mapa de caminos intenta construir un grafico de caminos, por los que es seguro viajar entre dos obstáculos. Una vez que se tenga el grafico disponible, el movimiento se calcula desplazando al robot desde la configuración inicial hasta alguna posición en un camino cercano, siguiendo después las diferentes rutas hasta acercarse a la

configuración final, y finalmente dejando la última ruta y moviendo el robot hasta la configuración final.

Éstos dos primeros métodos son llamados métodos globales: construyen una estructura de datos que es usada para encontrar los caminos entre dos configuraciones distintas del robot. Una ventaja de esto es que la estructura de datos sólo debe ser calculada una vez, y puede ser reutilizada para muchas configuraciones inicial final distintas. Sin embargo, estas estructuras de datos tienden a ser muy grandes, y los cálculos geométricos requeridos son complicados y de alto nivel de procesamiento.

Campos Potenciales.

Los métodos de campo potencial son más conocidos como métodos locales y forman una aproximación al problema radicalmente diferente. Estos métodos realizan el calculo del movimiento para una configuración inicial-final dada, parten del punto inicial e intentan guiar el robot en pequeños pasos hacia el objetivo final. La dirección de estos pasos esta determinada por una "fuerza". Se asume que el punto final o de destino produce una fuerza de atracción en el robot y los obstáculos una fuerza repulsiva. El robot es de esta forma atraído hacia el objetivo mientras es repelido por los obstáculos, evitando así las colisiones. El principal inconveniente de estos métodos es que en su forma más básica para problemas no triviales, a menudo conducen al robot hacia un *mínimo local* en donde las diferentes fuerzas se cancelan entre si por lo que el robot tiende a detenerse sin alcanzar su destino.

5.3 Algoritmo de Navegación Desarrollado

Hasta ahora ya se han descrito todos los elementos necesarios para un sistema de navegación:

- Unidad de control (procesamiento)
- Subsistemas para desplazamiento
- Marco de referencia de posicionamiento
- Mecanismo para posicionamiento

Por lo que se describirá el algoritmo de navegación desarrollado para este proyecto, como primer paso se intentó adaptar uno de los métodos de planificación de trayectorias descritos en la sección anterior. Como primera conclusión se tuvo que los métodos globales no pueden ser aplicables a este proyecto puesto que las condiciones iniciales son distintas a las que se tienen en el escenario de este proyecto. Esto es debido a que los métodos globales parten del punto en el que el espacio configuración del móvil puede modelarse en base a las distintas configuraciones del móvil y a la posición de los obstáculos. Lo que supone que el espacio de configuración del robot es estable como para mantener la estructura del modelo por los que se presupone que cuando el robot este detenido es decir que no produzca señales de control se mantenga en una sola posición sin cambio alguno, por lo que estos métodos no consideran la posibilidad de alteraciones al espacio de configuración provocadas por el medio lo que en el caso de este proyecto es imposible ya que se propone un sistema de control para un vehículo aéreo de desplazamientos a baja velocidad en campo abierto.

Para el caso específico del sistema de control propuesto en este proyecto con una gran masa que carece de una gran capacidad aerodinámica debido a su forma, el medio en el que éste se desplazará es claramente susceptible a cambios, sobre todo en lo que se refiere a la posición debido al viento, lo que prácticamente imposibilita a utilizar un método global que depende de la estabilidad del móvil.

Así mismo se estudio la posibilidad de utilizar el método de los campos potenciales dando como resultado que este método permite una gran distorsión en la configuración del móvil puesto que siempre esta realizando actualizaciones de posición y orientación del mismo, para así generar la señales de control necesarias para realizar las correcciones en el movimiento del móvil, por lo que se adapta al mecanismo de posicionamiento utilizado en este proyecto, que por medio de un receptor GPS el móvil conoce su ubicación exacta, además de que por las mismas características de los receptores GPS actualiza los datos de posición constantemente. Por todas estas características se decidió utilizar un algoritmo del tipo de campos potenciales. Sin embargo hay que notar que la mayor virtud de este método se encuentra cuando hay obstáculos por lo que se desperdicia en el caso particular de este proyecto en el que no esta previsto un escenario con obstáculos ya que el vehículo navega a campo abierto.

De esta forma se concluyó que no es posible el diseño de un algoritmo basado estrictamente en el método de planificación de trayectorias de campos potenciales, por lo que se adaptó de una forma muy práctica, dándole al algoritmo de navegación un enfoque local lo que representa un análisis de "cinemática a la inversa" del móvil, esto quiere decir que partiendo de que se conoce el

destino final que se pretende alcanzar, los movimientos del sistema se calculan a partir de este punto final, al contrario de la "cinemática directa" que analiza el movimiento del móvil, calculando las variables del sistema para conocer el punto de destino o final del mismo.

A continuación se presenta la explicación del algoritmo de navegación que se diseñó

5.3.1 Estructura del Algoritmo de Navegación

La primer parte del algoritmo consiste en determinar la posición del vehículo en un espacio dado. Para obtener la posición comenzamos con el mecanismo de posicionamiento que como ya se ha mencionado antes consta de un receptor GPS que proporciona varios datos acerca de la posición. Ya que el móvil se desplaza en tres dimensiones su posición esta dada por tres coordenadas (X,Y,Z) latitud, longitud y altitud, estas coordenadas las proporciona el Receptor GPS a razón de una vez por segundo. A partir de que se obtuvieron las coordenadas del móvil se calculan los ángulos α y β de rumbo y dirección respectivamente, el ángulo β o de dirección es el que se forma con la horizontal (eje x) o eje de las longitudes y la recta que pasa por los puntos definidos por la posición anterior y la posición instantánea, este ángulo nos indica la dirección en la que el vehículo se ha movido.

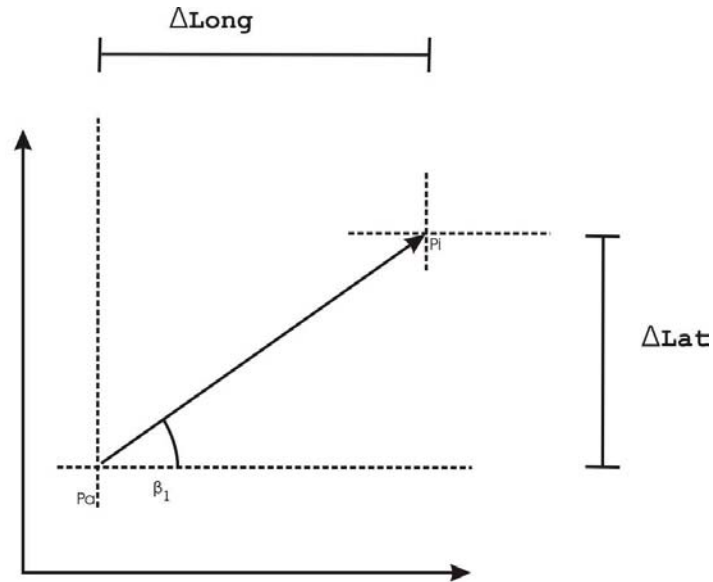


Figura 5.1 representación del ángulo β o de dirección

Este ángulo se obtiene cada segundo cuando se toma una nueva lectura del receptor GPS, por lo que la posición instantánea sustituye a la posición anterior cada segundo cuando se obtiene una nueva lectura. Esto es al instante t_0 la posición instantánea p_i es la posición inicial y por lo tanto también la posición anterior p_a , para el instante t_1 la posición recién adquirida es la posición instantánea P_i y la posición inicial es la posición anterior p_a , para el instante t_2 la posición que en el instante t_1 era la p_i pasa a ser la posición anterior p_a , y la posición recién adquirida a p_i y así sucesivamente cada segundo cuando se tiene una nueva lectura GPS, así mismo cada vez que se actualizan las posiciones también se calcula el ángulo β o de dirección, por lo que se tiene un ángulo cada segundo.

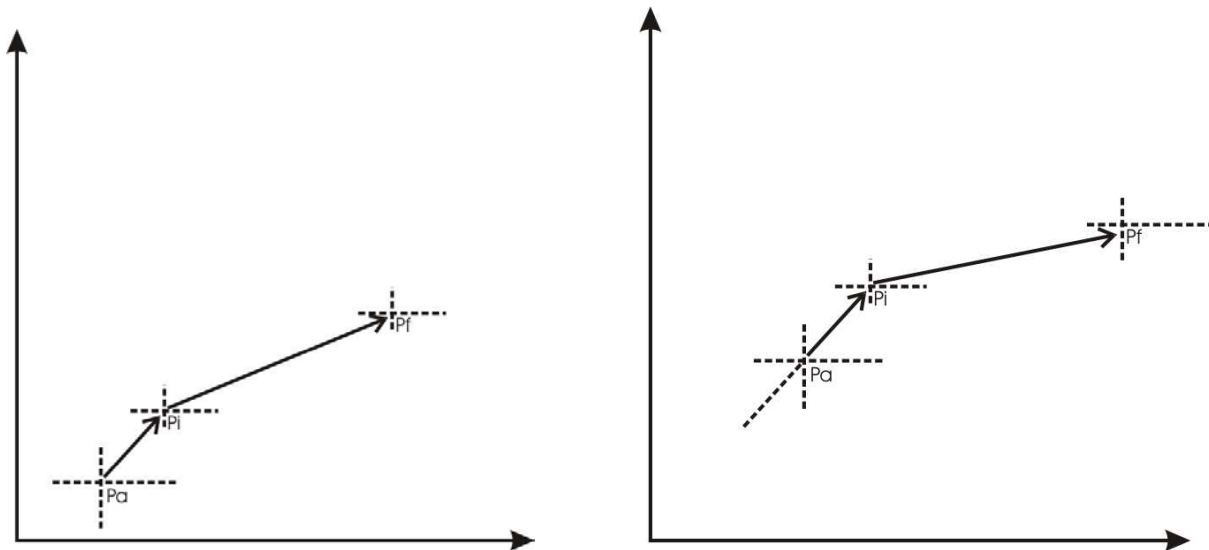


Figura 5.2 Actualización de coordenadas de posición

El uso de una posición instantánea y una anterior parte de que se debe tomar en cuenta que para la obtención de una configuración del vehículo son necesarias dos lecturas GPS en posiciones distintas, pues el sistema GPS no aporta datos acerca de la orientación del móvil. Por lo que en la primera lectura del receptor GPS no puede calcularse el ángulo β o de dirección, puesto que no existe lectura anterior en ese instante, de forma que se espera y requiere de una segunda lectura para calcular la trayectoria y la configuración del móvil por primera vez.

Para calcular el ángulo β o de dirección se procede de la siguiente manera: a partir de que se tiene p_i y p_a se calculan las diferencias entre las latitudes (ΔLat) y de la longitudes (ΔLong) de las posiciones instantánea y anterior, una vez que se tienen calculadas las diferencias tanto de latitud como de longitud se procede a realizar el cociente entre ellas para de esta forma obtener la tangente del ángulo β o de dirección, el cociente se realiza dividiendo

$$\frac{\Delta Lat}{\Delta Long} = \tan \beta \quad 5.1$$

donde ΔLat representa el cateto opuesto y $\Delta Long$ representa el cateto adyacente en un triángulo rectángulo que se forma entre ΔLat , $\Delta Long$ y la recta que pasa por las posiciones inicial y anterior.

La siguiente parte del algoritmo consiste en calcular el ángulo α o ángulo de rumbo, que se forma entre la horizontal (eje x) o el eje de las longitudes y la recta que pasa por los puntos definidos por las coordenadas de la posiciones final o de destino y la posición actual o instantánea del vehículo en cada lectura del receptor GPS, el ángulo α o de rumbo representa el rumbo que el vehículo debe seguir para alcanzar la posición final o de destino.

El calculo de este ángulo se hace de manera muy similar al del ángulo β o de dirección solo que se toman en cuenta p_i y p_f para calcular las diferencias de latitudes y de longitudes ΔLat y $\Delta Long$ para después calcular la tangente de forma similar que como se hizo con el ángulo β o de dirección, y al igual que este se realiza el cálculo cada segundo cuando se adquiere una nueva lectura GPS.

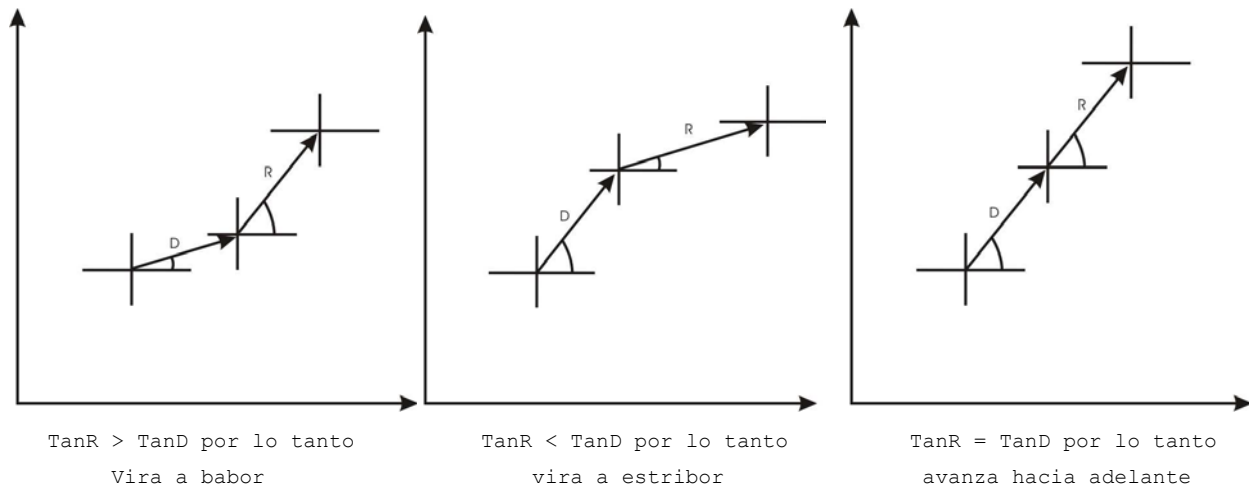


Figura 5.4 relación entre el ángulo α de rumbo y el ángulo β o de dirección

Para cada uno de los ángulos se pueden presentar cuatro casos dependiendo del cuadrante en el que se encuentre, para determinar el cuadrante se utilizara los signos de las diferencias de longitud y latitud ΔLatR y ΔLongR para el ángulo α o de rumbo y ΔLatD y ΔLongD para el ángulo β de dirección a continuación se presenta un ejemplo de los cuatro caso en los que se puede encontrar el ángulo α

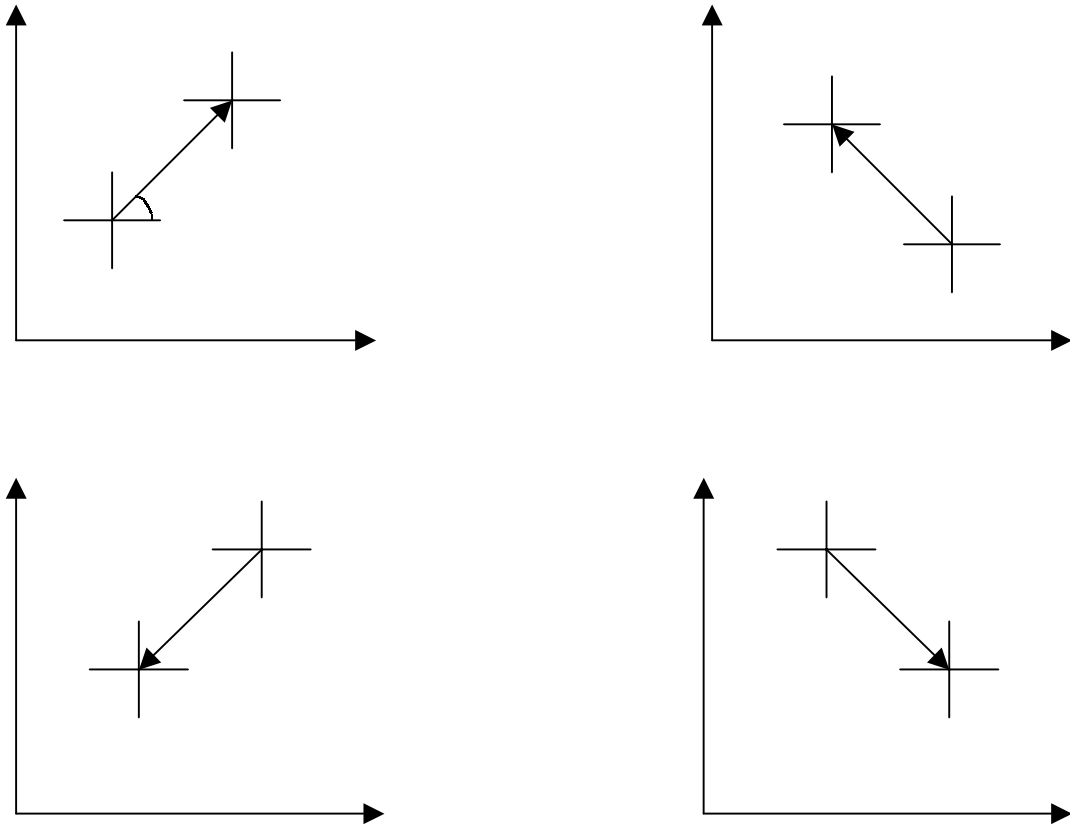


Figura 5.5 cuatro caso en los que se presenta el ángulo α

Como se puede inferir ya que tenemos dos ángulos combinados y cada uno de estos ángulos tiene 4 casos particulares partiendo del cuadrante en el que se encuentre el ángulo tenemos 16 combinaciones posible en las cuales el móvil se puede encontrar.

Para clasificar la acción que debe de tomar el sistema, el algoritmo identifica el caso en el que se encuentra el vehículo, primero a partir de los signos de ΔLatR y ΔLongR del ángulo α de rumbo, clasifica la tangente de rumbo, obteniendo una primera clase y después con los signos de ΔLatD y ΔLongD del ángulo β de dirección clasificar la tangente de dirección, y así obtener el caso en específico en el que se encuentra, para de esta forma tomar la decisión del movimiento del vehículo, a continuación se presenta cada uno de los 16 casos en específico que se pueden

presentar en este proyecto y como se clasifican dependiendo del cuadrante en el que se encuentra cada uno de sus ángulos .

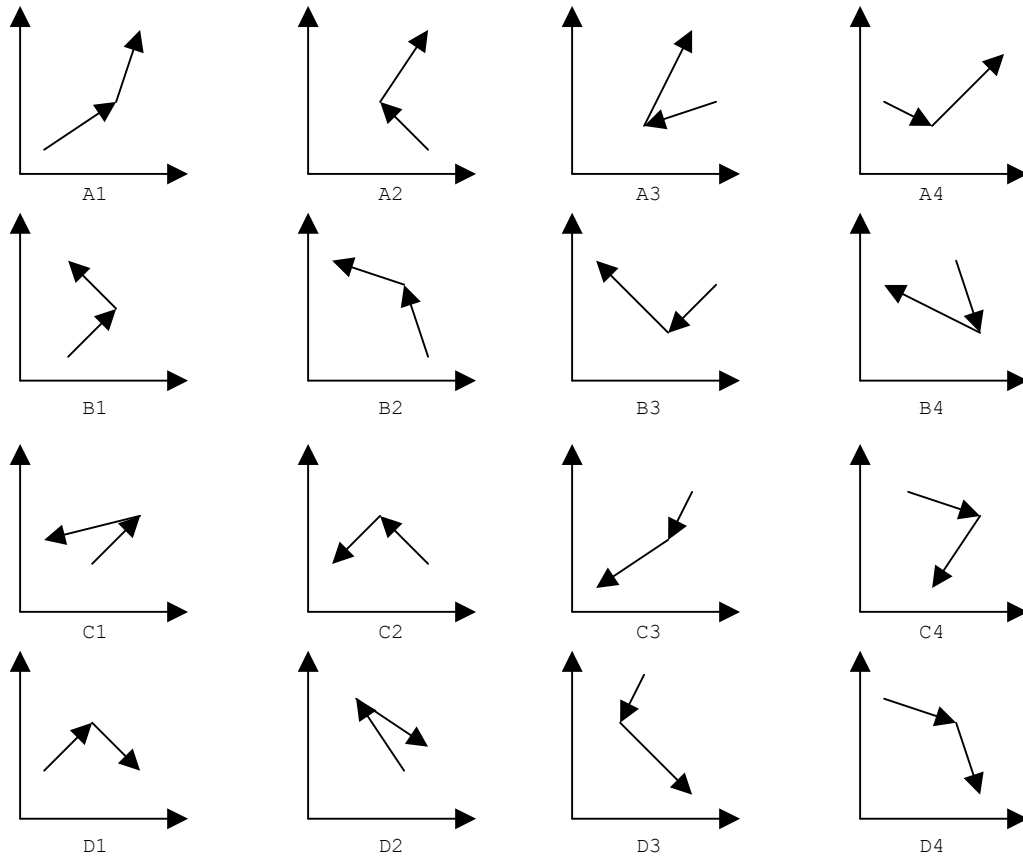


Figura 5.6 casos en particular dependiendo del cuadrante de en el que se encuentren los ángulos α y β

Todos los casos anteriores se pueden relacionar en una tabla la cual se presentara un poco mas adelante puesto que hay que contemplar otras situaciones; el algoritmo dependiendo del caso en el que se encuentre puede realizar 4 acciones que son las siguientes: izquierda, derecha, decisión 1 y decisión 2 figura 5.7. Ambas decisiones analizan las tangentes de dirección y de rumbo comparando una con otra de manera que para la decisión 1 se tiene la siguiente regla: si $\text{TanR} < \text{TanD}$ = derecha y si $\text{TanR} > \text{TanD}$ = izquierda. La decisión 2 tiene la siguiente regla: si $\text{TanR} < \text{TanD}$ = izquierda y si $\text{TanR} > \text{TanD}$ = derecha. Estas decisiones

son inversas una de la otra y se presentan cuando los dos ángulos se presentan en el mismo cuadrante. La decisión 1 se aplica en el caso en el que ambos ángulos se presentan en el cuadrante 1 y 3, la decisión 2 se aplica en el caso en el que ambos ángulos se presenten en el cuadrante 2 y 4 figura 5.8, por lo que podemos apreciar que se aplicara la misma decisión a cuadrantes opuestos por el origen.

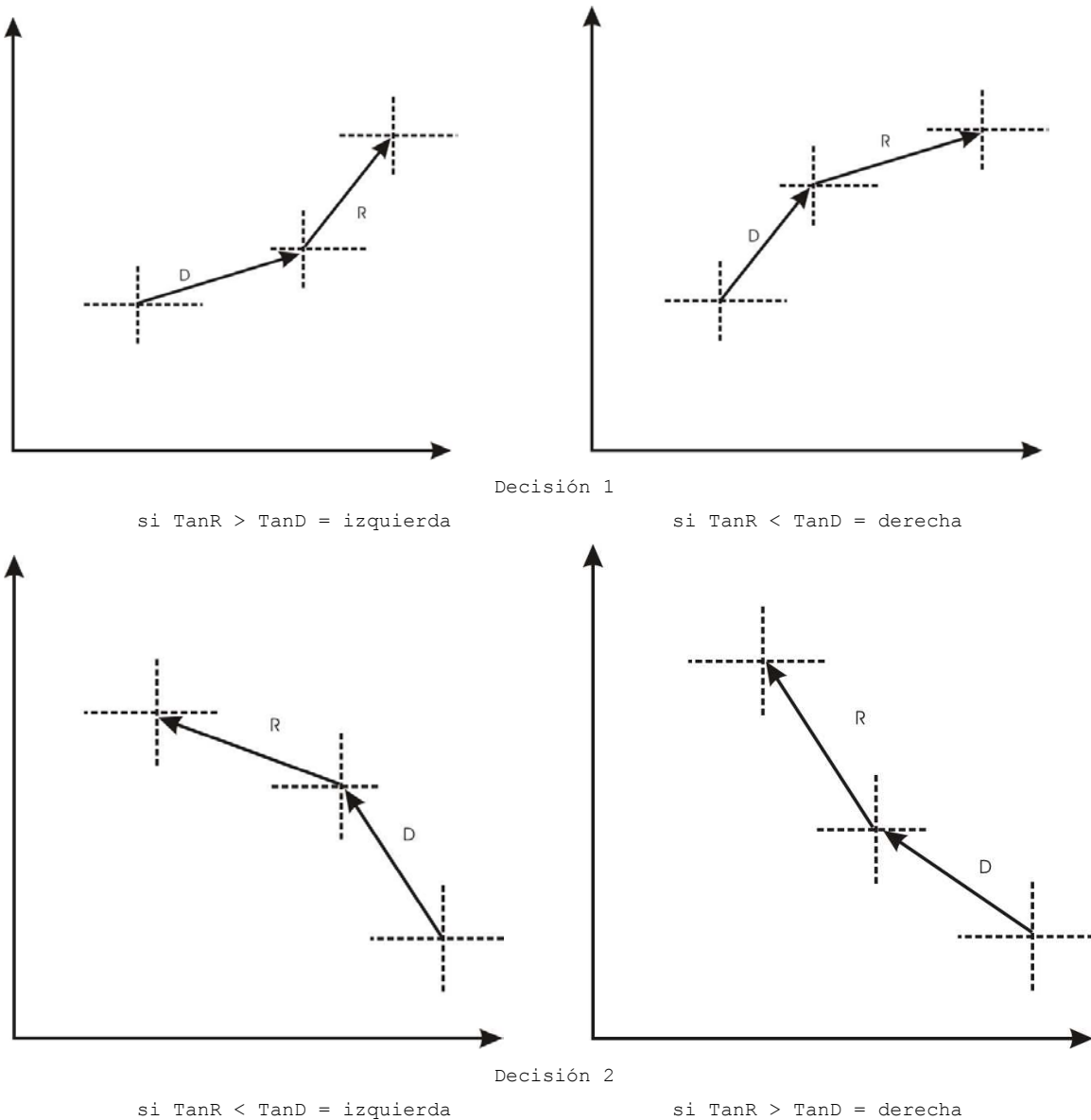


Figura 5.7 decisiones 1 y 2

ejemplos de los 32 casos en los que se puede presentar un componente paralelo a los ejes.

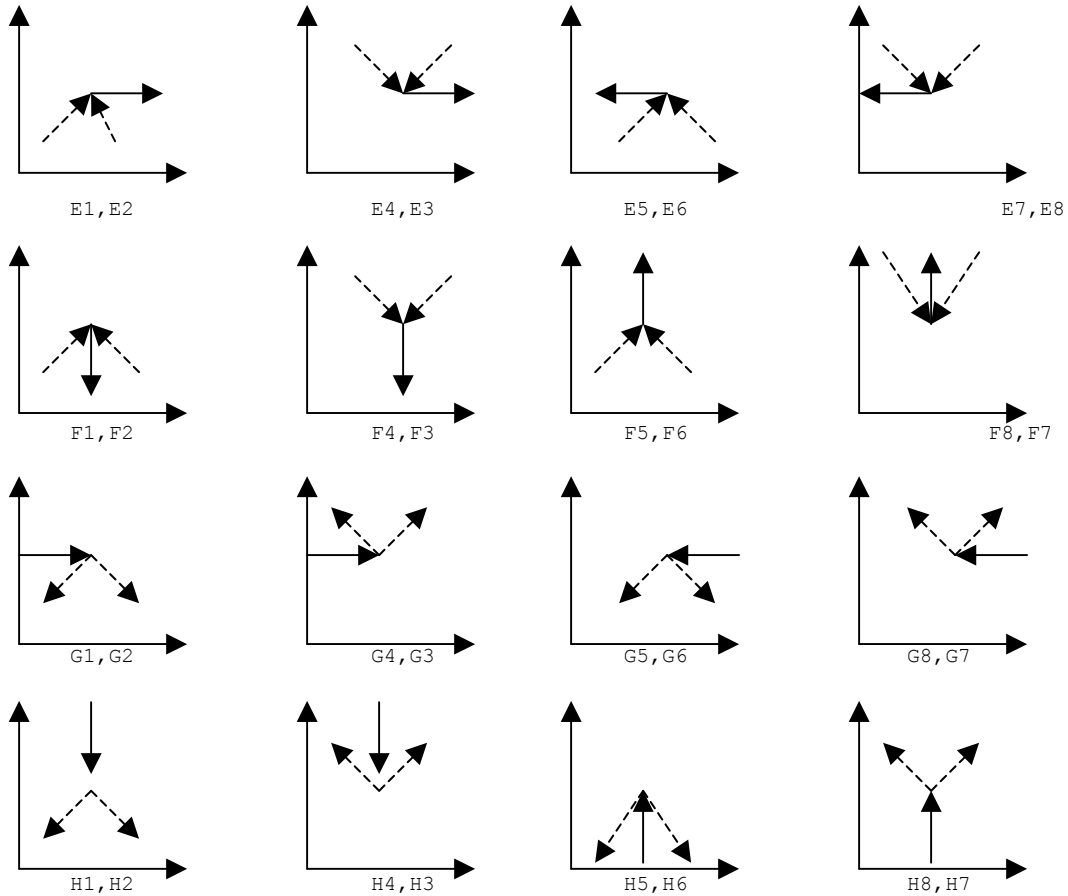


Figura 5.9 casos en particular en los que se presenta una componente paralela a los ejes

La clasificación se toma a partir de cual de las variables o componentes (ΔLatR , ΔLongR , ΔLatD y ΔLongD) tiene un valor 0 separándolo en cuatro diferentes ramas y dependiendo del signo de las otras 3 variables se determina el caso específico en el que se encuentra y a partir de aquí realizar una acción de manera muy similar a lo que se hace para los casos anteriores donde todas las variables tienen un valor, Estos casos también se relacionan en una tabla la cual se presentara un poco mas adelante. Como resultado de esta clasificación se pueden tomar 2 acciones

solamente: izquierda y derecha ya que estos casos se consideraran como estados intermedios antes de pasar a un caso el cual lleve a el rumbo correcto del vehículo

Así como se analizaron los casos en los que se puede presentar una variable con valor de cero también es necesario para los casos cuando se presenten dos variables con el valor de 0 o lo que es lo mismo con dos componentes paralelas a los ejes, figura 5.10, de manera similar al análisis para los casos donde solo una variable presenta un valor 0, primero se identifica cuales de las variables son las que presentan valores 0, aquí cabe recordar que solo se pueden presentar valores 0 en una de las componentes del ángulo α y una en las componentes del ángulo β al mismo tiempo puesto que si se presentan valores 0 para las 2 componentes de cualquiera de los dos ángulos o tres variables con valor 0 significa que el vehículo no se ha movido por lo que debe de existir alguna falla no relacionada con la unidad de control, una vez que se tiene identificados cuales de las variables tiene valores 0 se hace un análisis de los signos de las variables con valor diferente de 0 ya que estos signos nos dan la dirección en la que se ha movido o en la que se tiene que mover el vehículo. Este análisis nos da como resultado 16 posibles casos en los cuales puede encontrarse el móvil a un momento dado, por lo que tenemos ahora 64 posibilidades, al igual que cuando se presenta un solo valor 0 solo se pueden tomar 2 acciones izquierda y derecha puesto que también se consideran como estados intermedios, y también se relacionan en una tabla mas adelante.

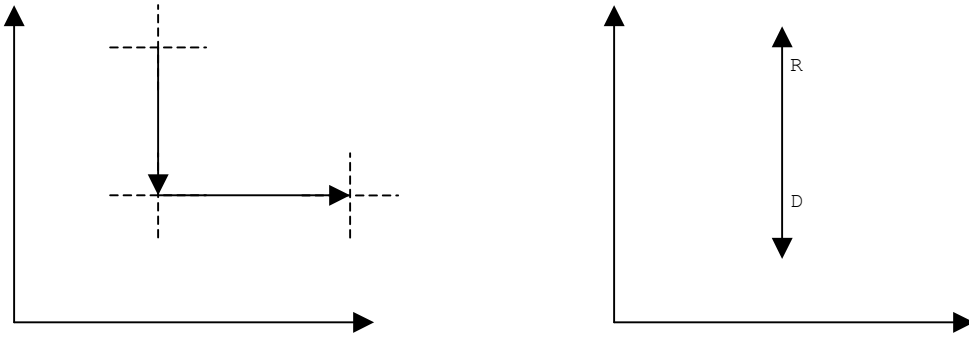


Figura 5.10 casos en los que se presentan dos componentes con valores 0 o paralelas a los ejes

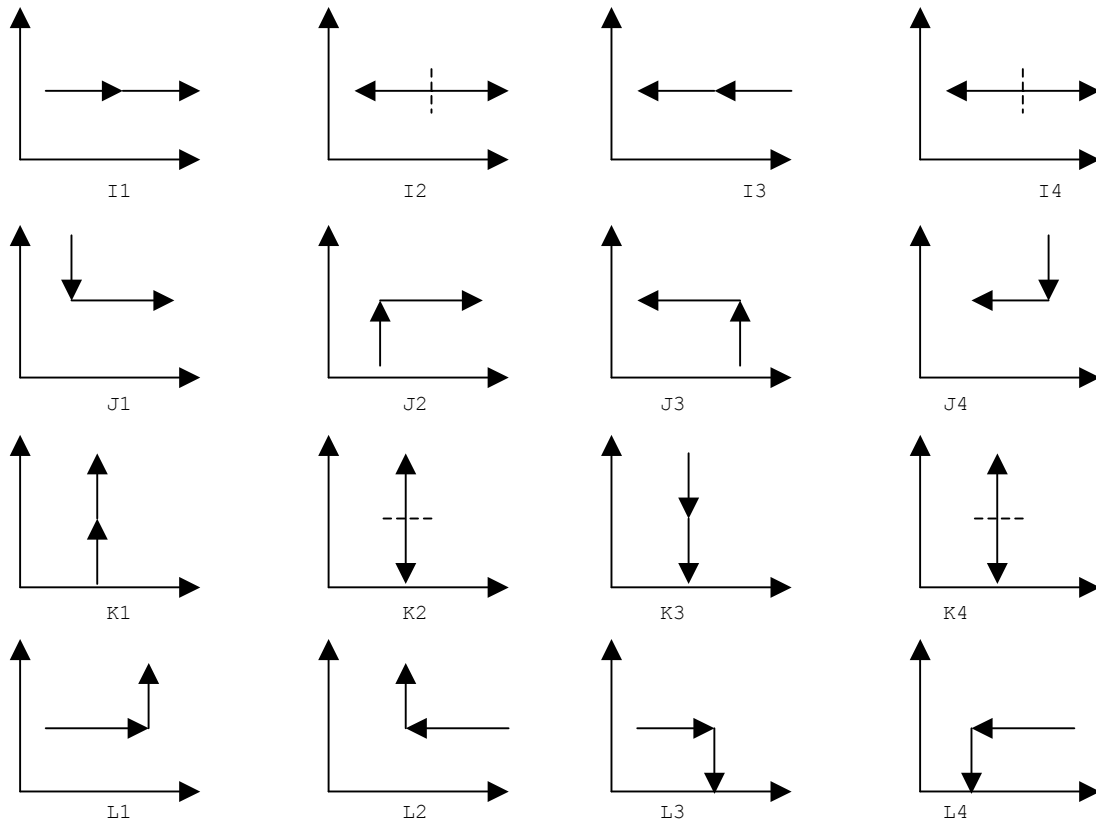


Figura 5.11 casos en los que se presenta dos variables con el valor 0

A continuación se presenta la tabla completa de los 64 casos que se pueden presentar en un momento dado

Tabla 5.1 clasificación de casos específicos que se pueden presentar para una configuración del móvil.

Numero	DLatR	DLongR	DLatD	DLongD	caso	Rutina	Simplificado
1	+	-	+	-	A1	Des1	A1
2	+	-	+	+	A2	Der	A2
3	+	-	-	+	A3	Des2	A3
4	+	-	-	-	A4	Izq	A4
5	+	+	+	-	B1	Izq	B1
6	+	+	+	+	B2	Des2	B2
7	+	+	-	+	B3	Der	B3
8	+	+	-	-	B4	Des1	B4
9	-	+	+	-	C1	Des2	C1
10	-	+	+	+	C2	Izq	C2
11	-	+	-	+	C3	Des1	C3
12	-	+	-	-	C4	Der	C4
13	-	-	+	-	D1	Der	D1
14	-	-	+	+	D2	Des1	D2
15	-	-	-	+	D3	Izq	D3
16	-	-	-	-	D4	Des2	D4
17	0	-	+	-	E1	Der	E1
18	0	-	+	+	E2	Der	E1
19	0	-	-	+	E3	Izq	E2
20	0	-	-	-	E4	Izq	E2
21	0	+	+	-	E5	Izq	E3
22	0	+	+	+	E6	Izq	E3
23	0	+	-	+	E7	Der	E4
24	0	+	-	-	E8	Der	E4
25	-	0	+	-	F1	Der	F1
26	-	0	+	+	F2	Izq	F2
27	-	0	-	+	F3	Izq	F2
28	-	0	-	-	F4	Der	F1
29	+	0	+	-	F5	Izq	F3
30	+	0	+	+	F6	Der	F4
31	+	0	-	+	F7	Der	F4
32	+	0	-	-	F8	Izq	F3
33	+	-	0	-	G1	Izq	G1
34	+	+	0	-	G2	Izq	G1
35	-	+	0	-	G3	Der	G2
36	-	-	0	-	G4	Der	G2
37	+	-	0	+	G5	Der	G3
38	+	+	0	+	G6	Der	G3
39	-	+	0	+	G7	Izq	G4
40	-	-	0	+	G8	Izq	G4

41	+	-	-	0	H1	Izq	H1
42	+	+	-	0	H2	Der	H2
43	-	+	-	0	H3	Der	H2
44	-	-	-	0	H4	Izq	H1
45	+	-	+	0	H5	Der	H3
46	+	+	+	0	H6	Izq	H4
47	-	+	+	0	H7	Izq	H4
48	-	-	+	0	H8	Der	H3
49	0	-	0	-	I1	Avanza	I1
50	0	-	0	+	I2	Der	I2
51	0	+	0	+	I3	Avanza	I3
52	0	+	0	-	I4	Der	I4
53	0	-	-	0	J1	Izq	J1
54	0	-	+	0	J2	Der	J2
55	0	+	+	0	J3	Izq	J3
56	0	+	-	0	J4	Der	J4
57	-	0	0	-	K1	Avanza	K1
58	-	0	0	+	K2	Der	K2
59	+	0	0	+	K3	Avanza	K3
60	+	0	0	-	K4	Der	K4
61	-	0	-	0	L1	Der	L1
62	-	0	+	0	L2	Izq	L2
63	+	0	+	0	L3	Der	L3
64	+	0	-	0	L4	Izq	L4

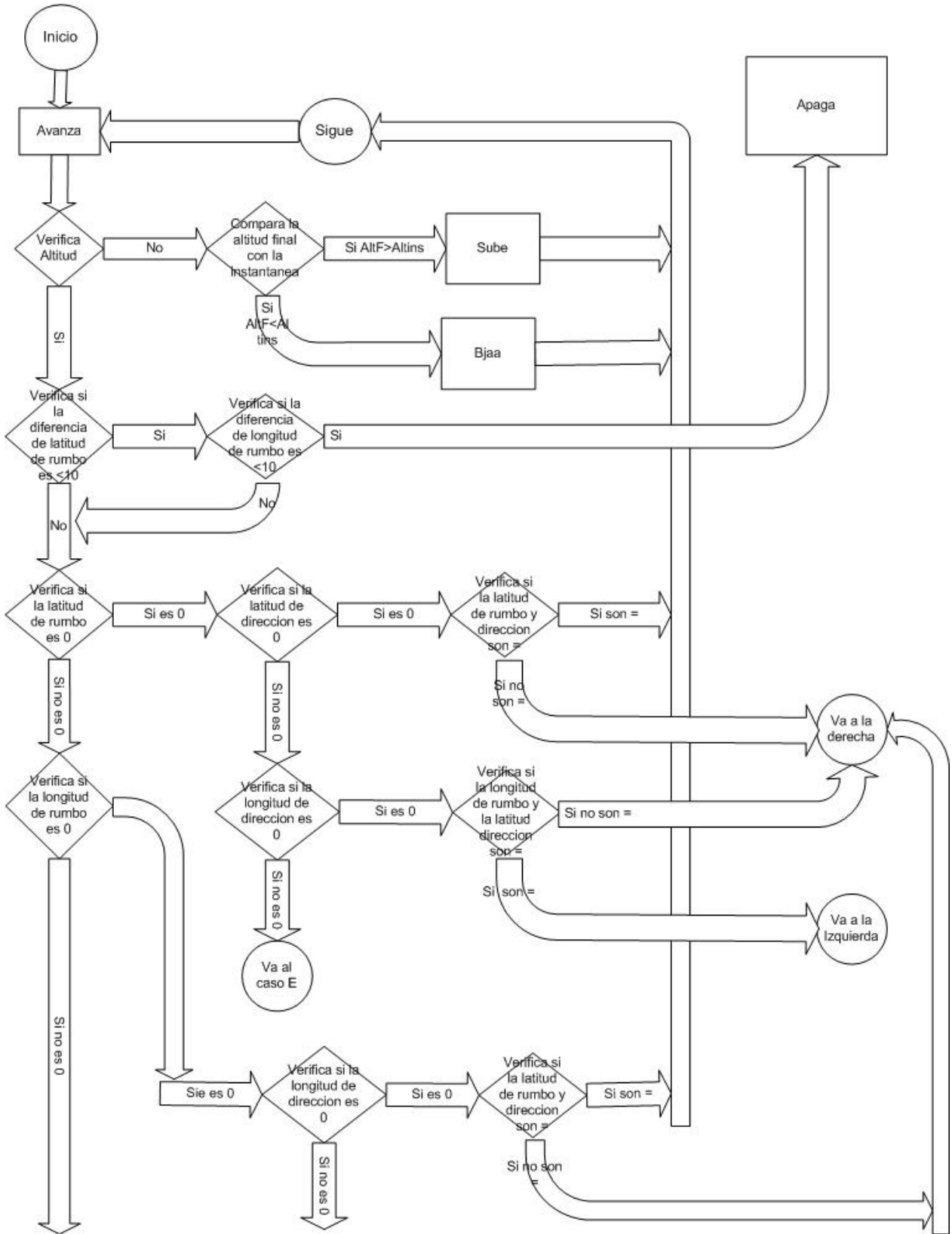
Por último haciendo un breve recuento de lo que hace el algoritmo tenemos los siguientes pasos en el algoritmo:

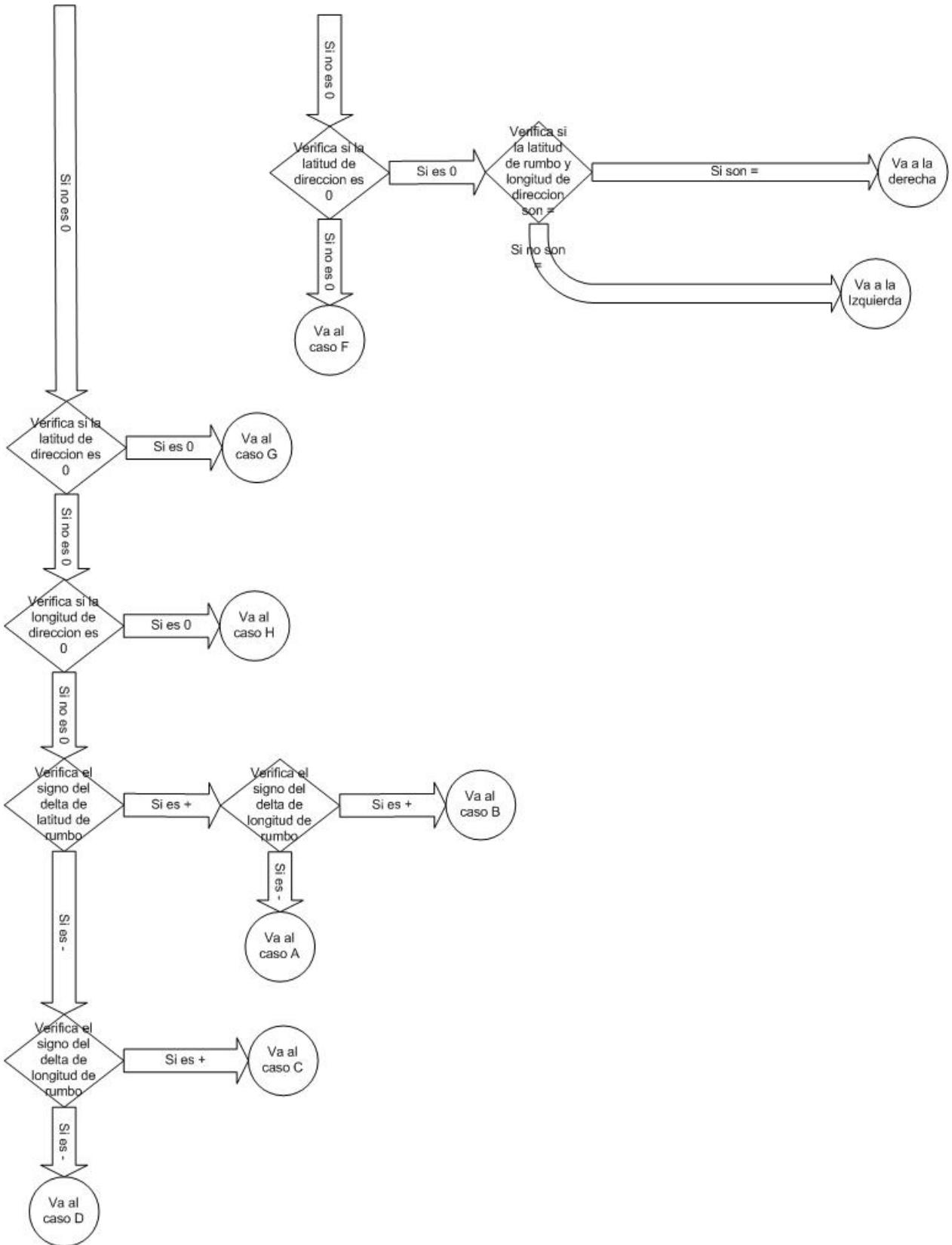
1. Después de haber iniciado avanza para de esta forma poder tener la primera lectura de dirección
2. verifica si la altitud del vehículo es igual a la del punto final, si no lo es, pasa a la rutina que eleva o desciende el vehículo, para después volver a comparar la altitud
3. verifica si la latitud y longitud son iguales a los del punto final, si lo son pasa a una rutina en la que apaga los motores y termina el algoritmo.
4. verifica si alguna de las componentes es 0, si encuentra dos componentes con valor cero, identifica

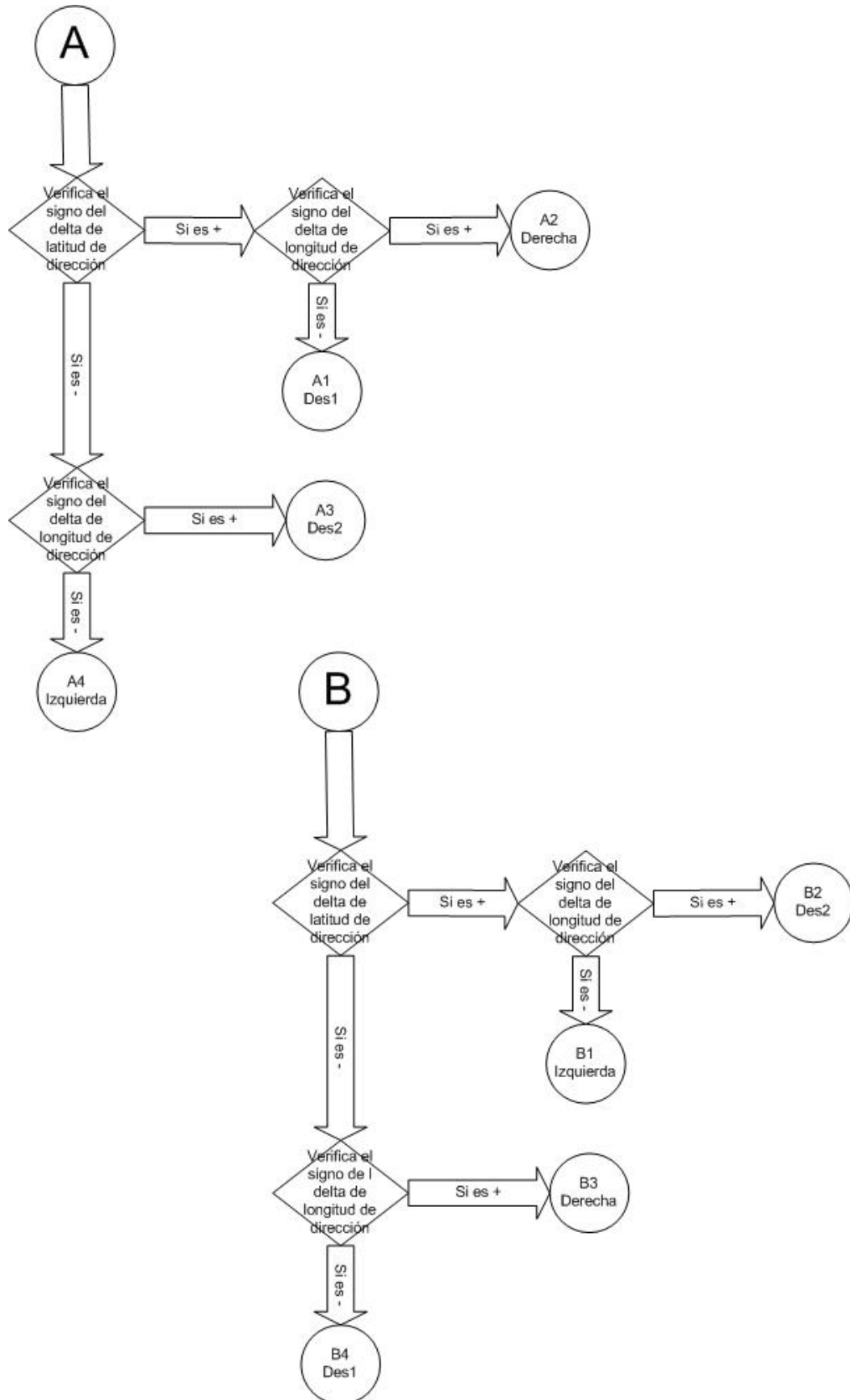
los signos de las componentes restantes para realizar la clasificación.

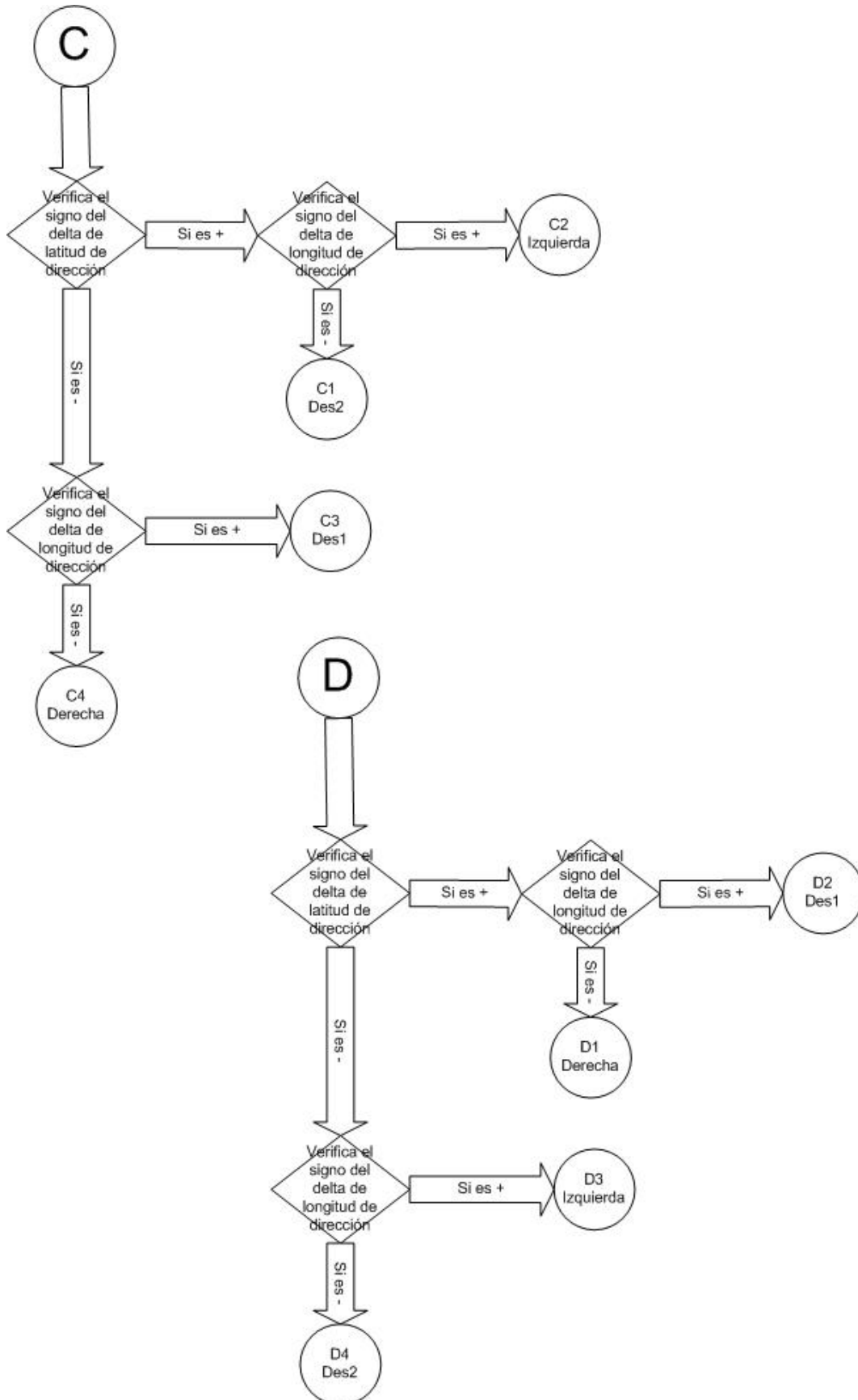
5. si solo una de las componentes es 0, identifica los signos de las componentes restantes para clasificarlas
6. si ninguna de las componentes es 0, identifica los signos de todas las componentes para clasificar el caso
7. una vez identificado el caso, dependiendo de este el programa realiza una acción en específico de movimiento para después pasar al punto 1 hasta que llega a su punto final

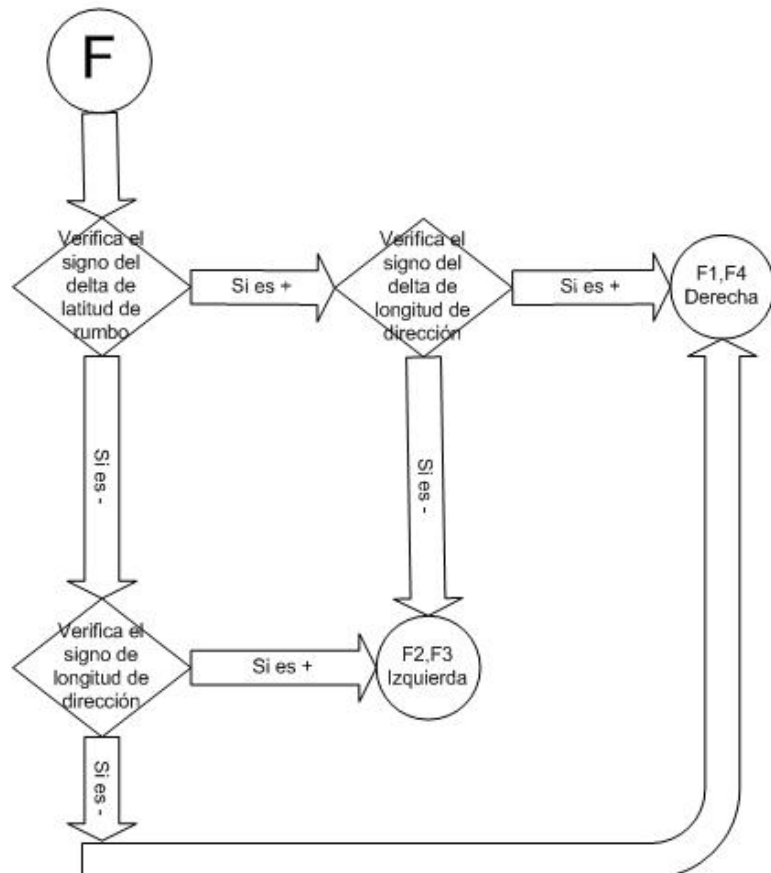
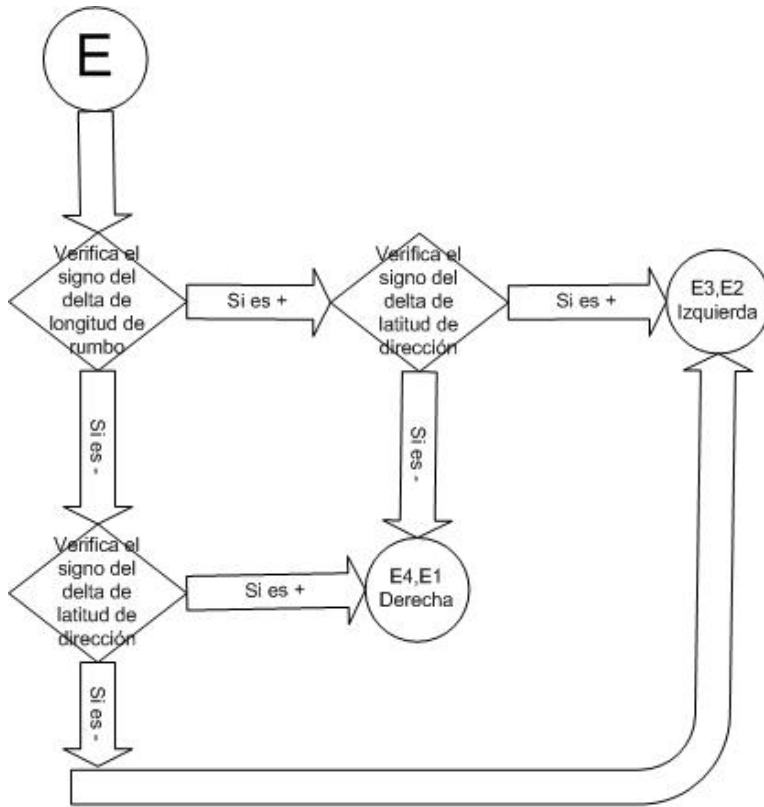
A continuación se presenta el diagrama de flujo del algoritmo

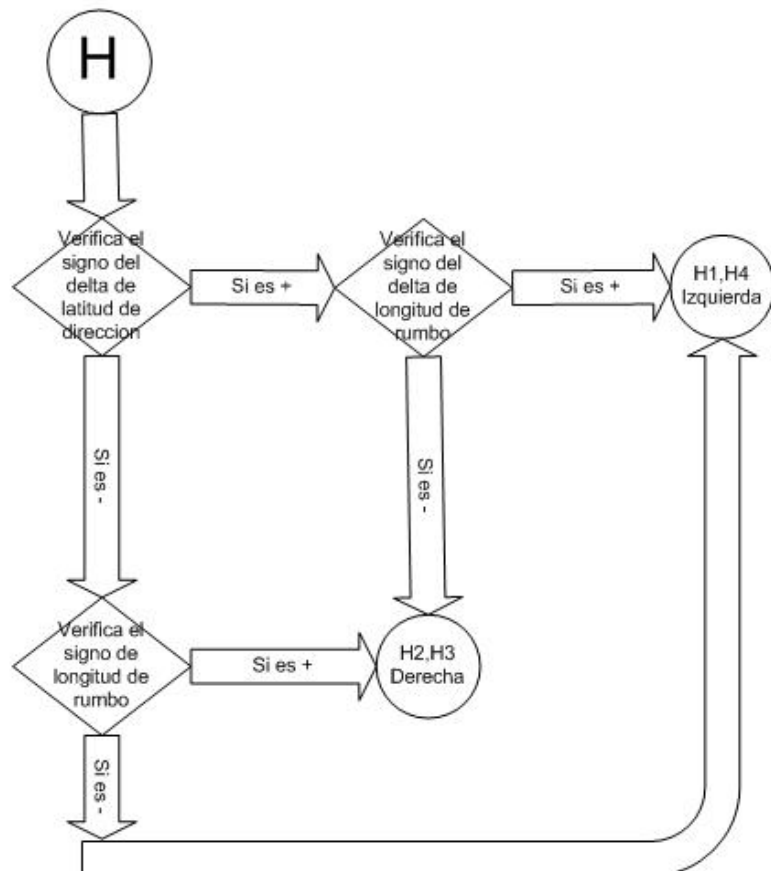
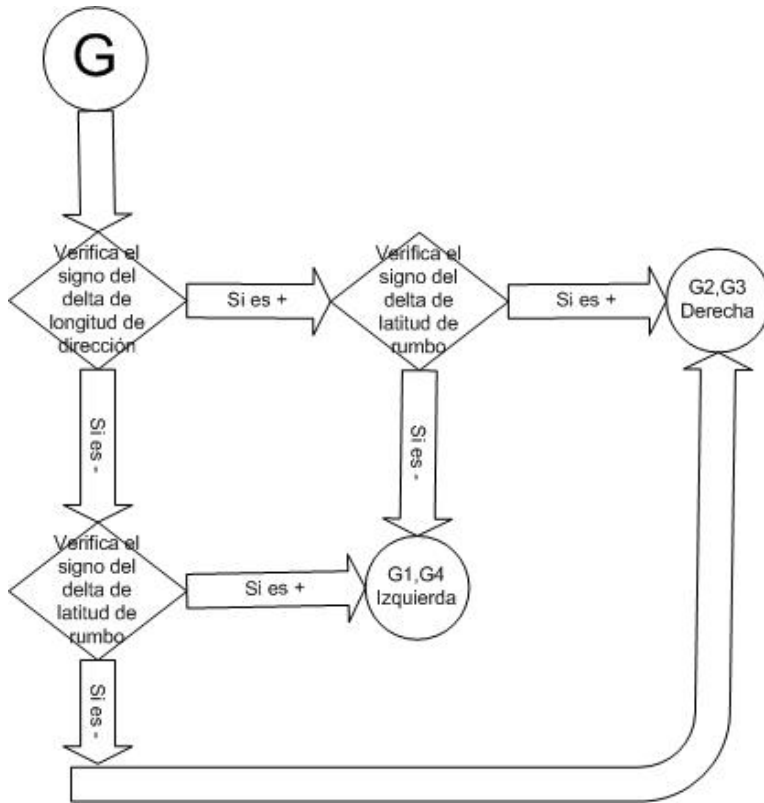












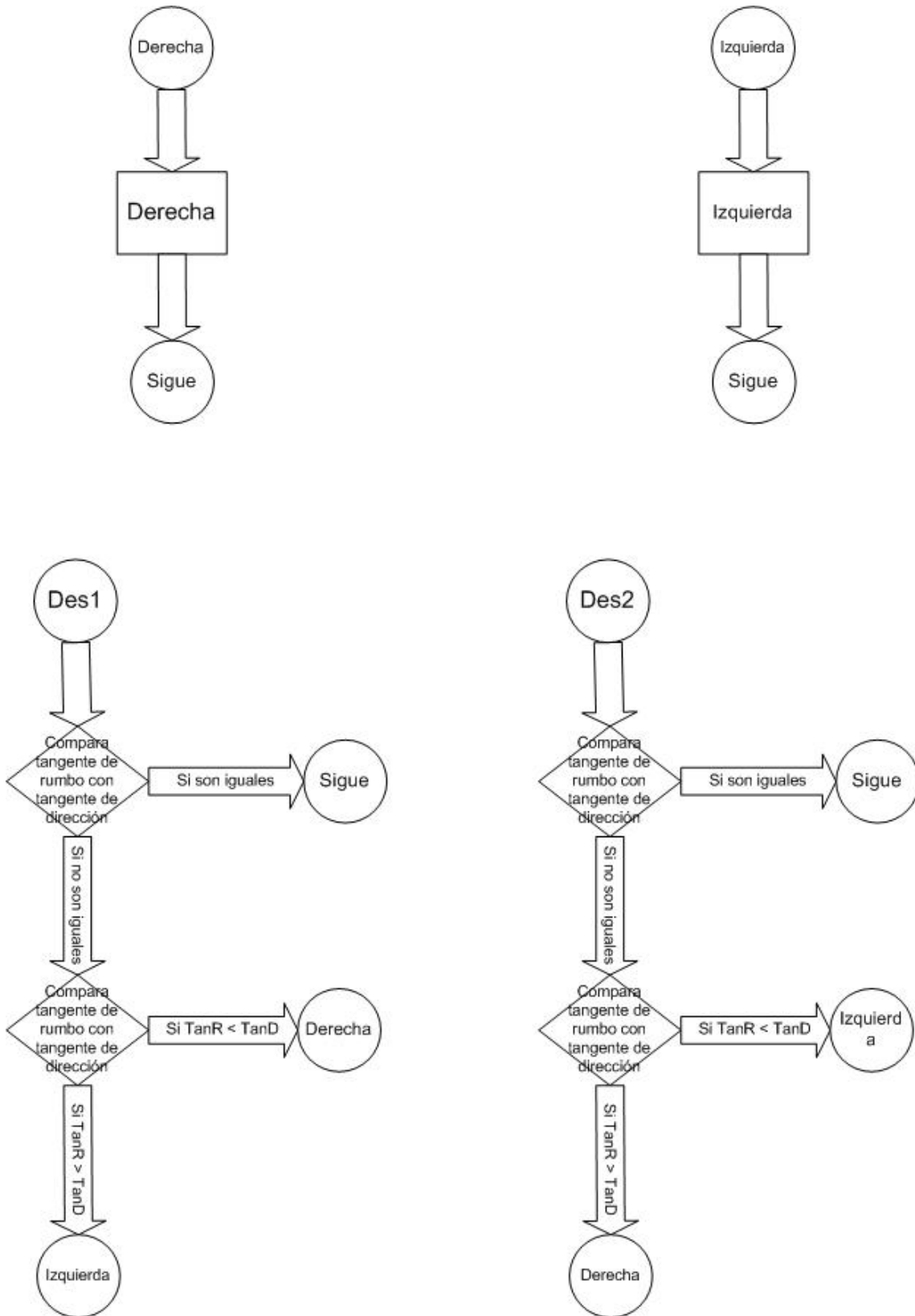


Figura 5.12 diagrama de flujo del algoritmo de navegación.

5.4 Referencias

World Geodetic System 1984

<http://www.wgs84.com/wgs84/wgs84.htm>

Guardiola, Eduardo; "Planificación de Trayectorias con rotaciones,
Robotica

Universidad de Alicante, España

ejgs@alu.ua.es

6. PRUEBAS Y RESULTADOS	1
6.1 PRUEBAS .	1
6.2 TIPOS DE PRUEBAS .	2
6.2.1 Pruebas de Altitud .	3
6.2.2 Pruebas de Posición .	4
6.2.3 Pruebas de Trayectoria .	9
6.2.4 Pruebas de Aproximación .	15

6. Pruebas y Resultados.

6.1 Pruebas.

En este capítulo se describen las pruebas que se llevaron a cabo al sistema autónomo de navegación. Las pruebas de altitud se desarrollaron en el Ajusco al sur de la ciudad de México para tener una variación de altitud mayor a los 50 metros que es el límite de aproximación establecido en el sistema. Las pruebas de dirección tuvieron lugar en los estacionamientos del estadio olímpico México 68 en ciudad universitaria ya que aquí se pueden encontrar espacios lo suficientemente grandes y sin obstáculos como para desplazar un móvil en prácticamente cualquier dirección.

Para realizar estas pruebas se suplió la etapa de potencia por un conjunto de displays de 7 segmentos donde, de esta forma se puede apreciar los comandos o códigos generados por la unidad de control a partir de las decisiones que esta toma dependiendo de la posición en la que se encuentra, y la dirección del punto al cual se dirige, de esta forma se puede seguir las instrucciones de navegación en automóvil. Al segundo par de displays se conectó la salida del control de altitud y la salida del control de avance, al tercer par de displays se conectó la salida del control de dirección. Figura 6.1

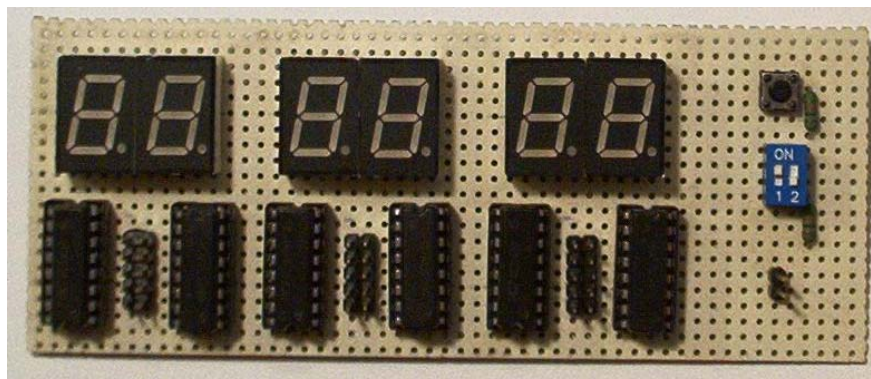


Figura 6.1 Conjunto de Displays

Los códigos generados por la unidad de control para generar el movimiento son:

Tabla 6.1 códigos de control

Subir	1, X, X, X
Bajar	2, X, X, X
Avanza	X, 1, X, X
Derecho	X, 1, 0, 0
Derecha	X, X, 0, 1
Izquierda	X, X, 0, 2
Parar	0, 0, 0, 0

Las pruebas se llevaron acabo con base en el siguiente protocolo: como primer paso se encendió el receptor GPS para que este se sincronice con el sistema NAVSTAR-GPS, una vez terminada esta sincronización se enciende la unidad de control y se introducen las coordenadas del punto de destino, a continuación se ingresa a la unidad de control la señal proporcionada por el receptor GPS, inmediatamente el sistema produce una instrucción la cual debe ser ignorada puesto que como las instrucciones dependen de una posición instantánea y una posición anterior se debe de mover el vehículo en cualquier dirección antes de tener una instrucción correcta a partir de que ya cuenta con 2 posiciones una instantánea y una anterior, una ves que se está en movimiento el sistema entrega instrucciones bastante confiables.

6.2 Tipos de pruebas.

Se diseñaron 4 tipos de pruebas: de altitud, de posición, de trayectoria y de aproximación. Las pruebas de altitud consistieron en alcanzar la altitud del punto final partiendo de una altitud mayor o menor diferente de la del punto destino para de esta manera seguir los comandos de altitud, las de posición consisten en dado un punto fijo seleccionado como de destino, identificar la dirección en la que se encuentra ese punto a partir de la posición del vehículo. Las pruebas de trayectoria consisten en, a partir de un punto cualquiera de inició desde el cual parte el vehículo se siguen los comandos generados por la unidad de control para alcanzar el punto de destino, y las pruebas de aproximación consisten en el acercamiento al punto de destino en línea recta partiendo desde diferentes direcciones.

6.2.1 Pruebas de Altitud.

Se realizaron 2 pruebas de altitud. La primera teniendo como punto de destino La entrada del parque de diversiones Six Flags en el Ajusco al sur de la ciudad de México con un altitud aproximada de 2440 m sobre el nivel del mar, y como punto de partida el estadio olímpico México 68 en la ciudad universitaria con una altitud aproximada de 2280 m sobre el nivel del mar.

Al iniciar el sistema y obtener los primeros datos del receptor GPS reportando la altitud de 2280 m la unidad de control entregó el siguiente código 1,X,X,X lo que significa que se tiene que ascender. Este calculo lo realizó a partir de la resta entre la altitud final y la altitud instantánea, y la comparación del resultado con el rango de aproximación de altura que es de 50 m.

El sistema mantuvo el mismo código hasta que se aproximó a unos 300 m del punto destino el cual esta por arriba de los 2400 m sobre el nivel del mar aproximadamente.

La segunda prueba tuvo lugar a la inversa se partió de la entrada del parque de diversiones Six Flgs con una altura aproximada de 2450 m sobre el nivel del mar y se ingresaron coordenadas del punto destino en el estacionamiento numero 8 del estadio Olímpico México 68 de la ciudad universitaria, y al igual que en la prueba anterior al recibir los primeros datos proporcionados por el receptor GPS la unidad de control generó el siguiente código 2,X,X,X el cual corresponde a descender.

6.2.2 Pruebas de Posición

Estas pruebas consisten en que a partir de un punto específico tomado como de referencia y cargado en el sistema como punto de destino, seguir diferentes trayectorias haciendo que trabaje el sistema y nos genere un código de dirección que podremos identificar como la dirección en la que se encuentra el punto de destino.

Como parte de estas pruebas se recolectaron todas las lecturas generadas por el receptor GPS y se incorporaron en un mapa de la zona, generando la traza de la trayectoria que siguió el móvil durante la prueba en el mapa, estas imágenes se muestran mas adelante en el texto.

Las pruebas de posición tuvieron lugar en el estacionamiento numero 8 del estadio olímpico México 68 de ciudad universitaria, Se colocó el punto N 19° 19.814 W 99° 11.451 con una altitud de 2290 m de altitud, mismo que corresponde al centro del estacionamiento y se ingresó como punto de destino en el sistema.

Alrededor de este punto se realizaron 2 círculos concéntricos uno a favor de las manecillas del reloj y el segundo en contra Figuras 6.2 y 6.3

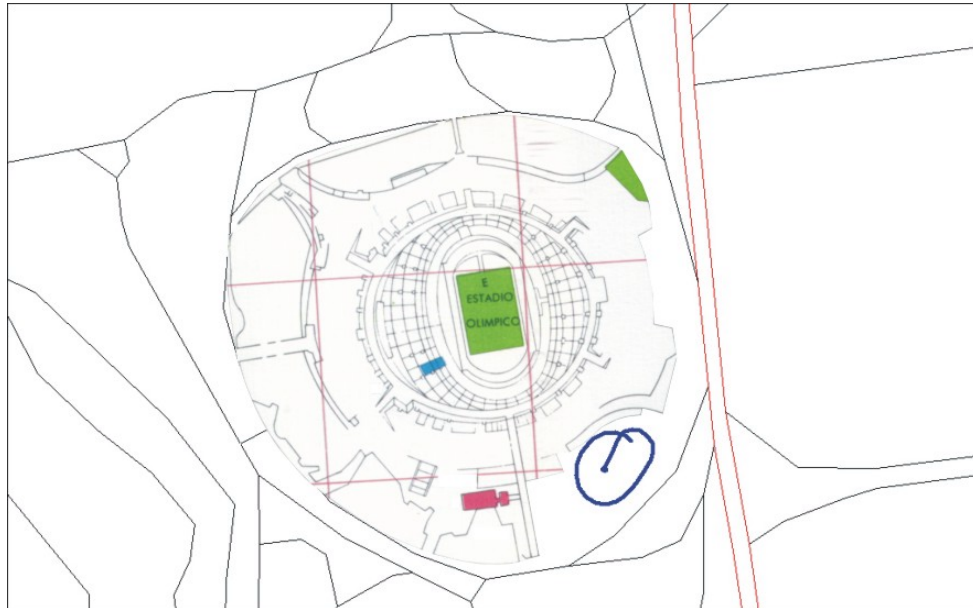


Figura 6.2 círculo a favor del reloj en el estacionamiento 8 del estadio olímpico

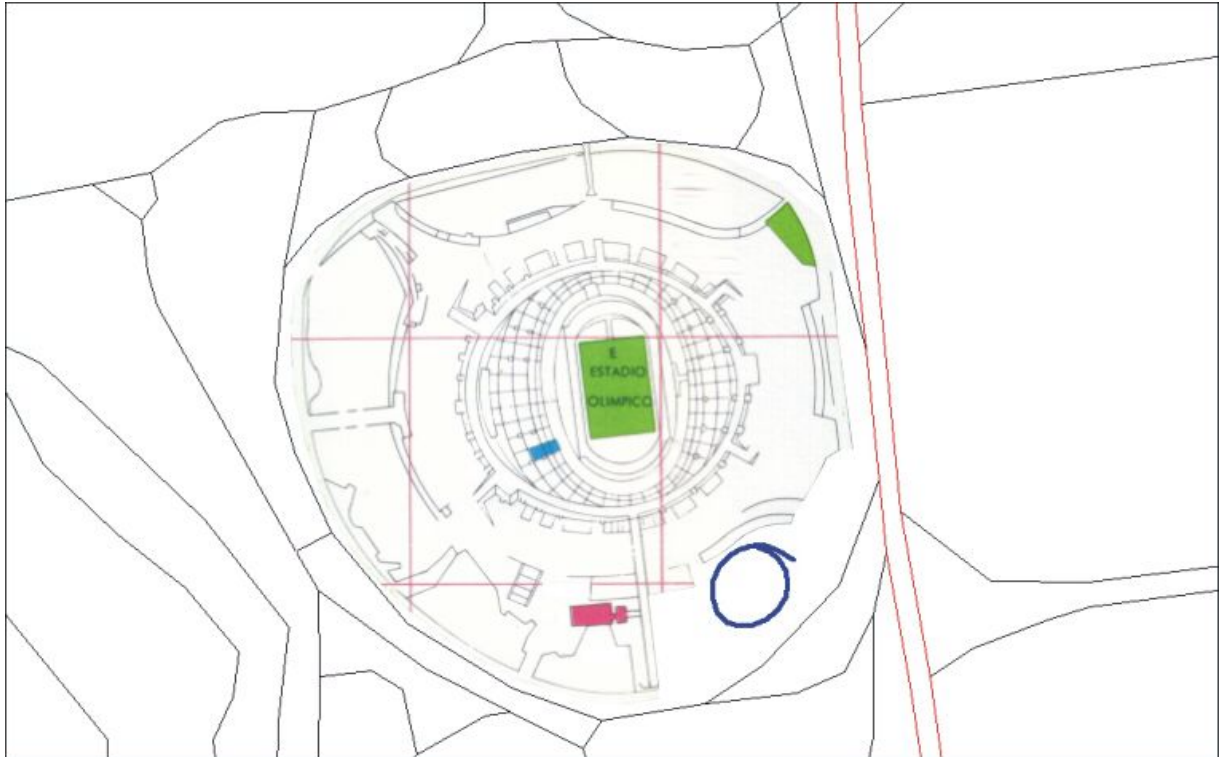


Figura 6.3 círculo a en contra del reloj en el estacionamiento 8 del estadio olímpico

La unidad de control generó el código $X,X,0,1$ indicando que el punto de destino se encontraba en todo momento a la derecha del móvil para el círculo con dirección a favor del reloj y el código $X,X,0,2$ indicando que el punto de destino se encontraba en todo momento a la izquierda del móvil para el círculo con dirección en contra del reloj, cada cálculo que generó estos códigos de dirección se realizaron a partir de tres posiciones, la instantánea, la anterior y la final en la figura 6.4 podemos apreciar los puntos que se captaron en las dos pruebas.

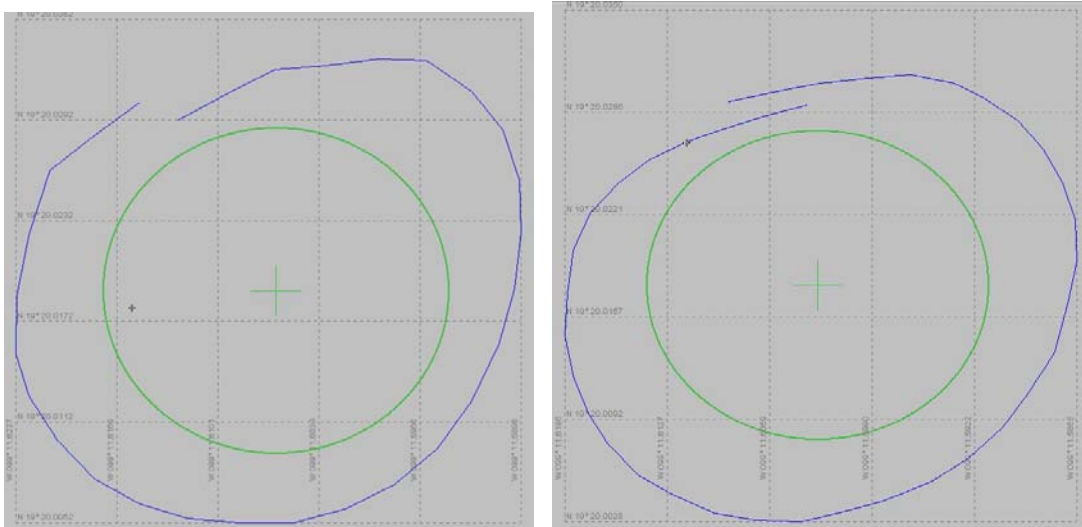


Figura 6.4 puntos captados las pruebas de posición

Las siguientes 2 pruebas también se realizaron en el estacionamiento numero 8 del estadio olímpico México 68 de ciudad universitaria, Se coloco el punto N 19° 19.803 W 99° 11.461 con una altitud de 2290 m de altitud, como punto de destino o de referencia y se ingresó como punto de destino en el sistema, se realizaron 2 recorridos generando una S en un sentido y el siguiente en sentido contrario. figuras 6.5 y 6.6

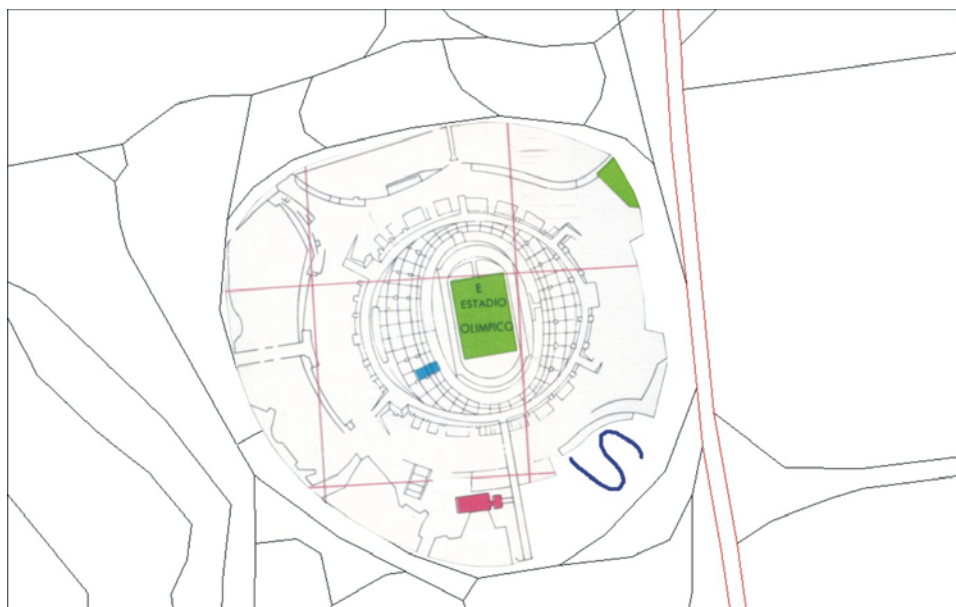


Figura 6.5 prueba de S en el estacionamiento 8 del estadio olímpico

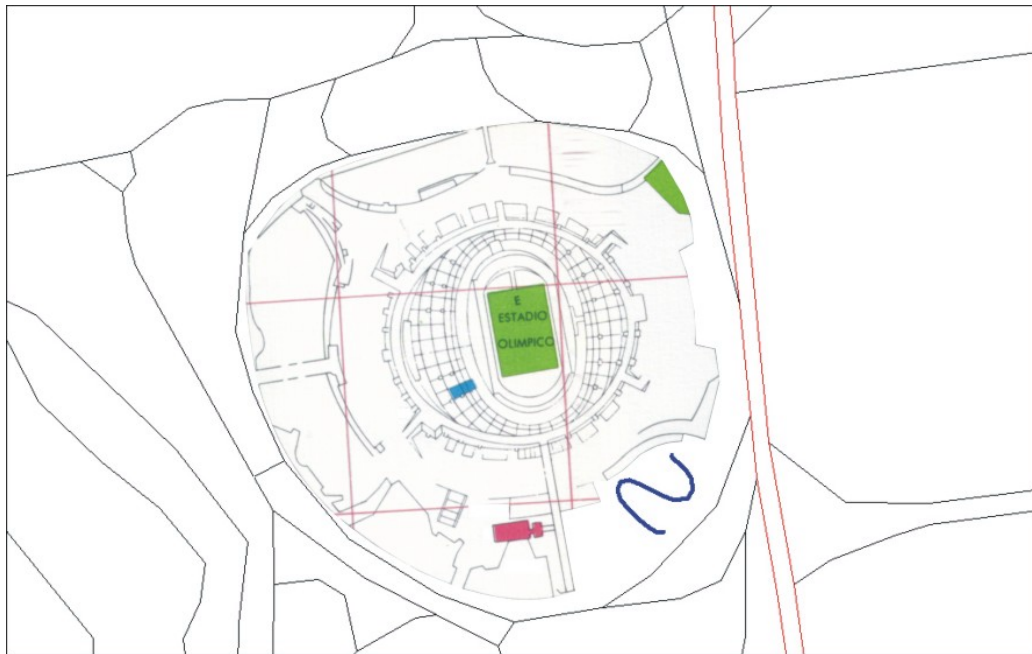


Figura 6.6 prueba de S invertida en el estacionamiento 8 del estadio olímpico

La unidad de control generó diferentes códigos para cada una de las pruebas pero siempre indicando la dirección en la que se encontraba el punto de referencia o de destino previamente cargado en el sistema, al igual que en las pruebas anteriores cada calculo que generó estos códigos de dirección se realizó a partir de tres posiciones, la instantánea y la anterior y la final en la figura 6.7 podemos apreciar los puntos que se captaron en las dos pruebas.

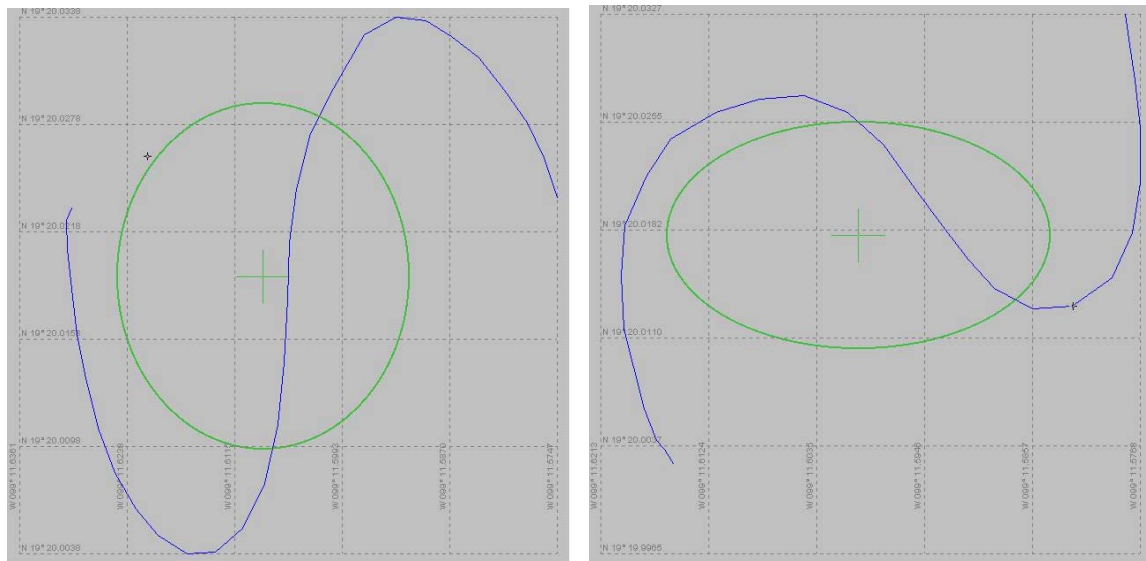


Figura 6.7 puntos captados las pruebas de posición

6.2.3 Pruebas de Trayectoria

Las pruebas de trayectorias consisten en que tomando un punto en el espacio de 2 dimensiones que interpretaremos como punto destino, lo cargaremos en el sistema como punto destino del vehículo y partiendo de otro punto cualquiera a una distancia aproximada de 250 m seguimos las instrucciones de dirección del sistema para llegar desde el punto de partida al punto final.

Las observaciones GPS generadas por el receptor para realizar estas pruebas fueron registradas e interpretadas gráficamente en un mapa de ciudad universitaria. Las imágenes generadas son presentadas mas adelante en esta sección.

Al igual que las pruebas de posición estas pruebas tuvieron lugar en el estacionamiento numero 8 del estadio olímpico México 68 de ciudad universitaria, Se colocó el punto N 19° 19.833 y W 99° 11.433 con una altitud de 2290 m de altitud mismo que se cargó en el sistema como punto destino. Se realizaron 8 pruebas

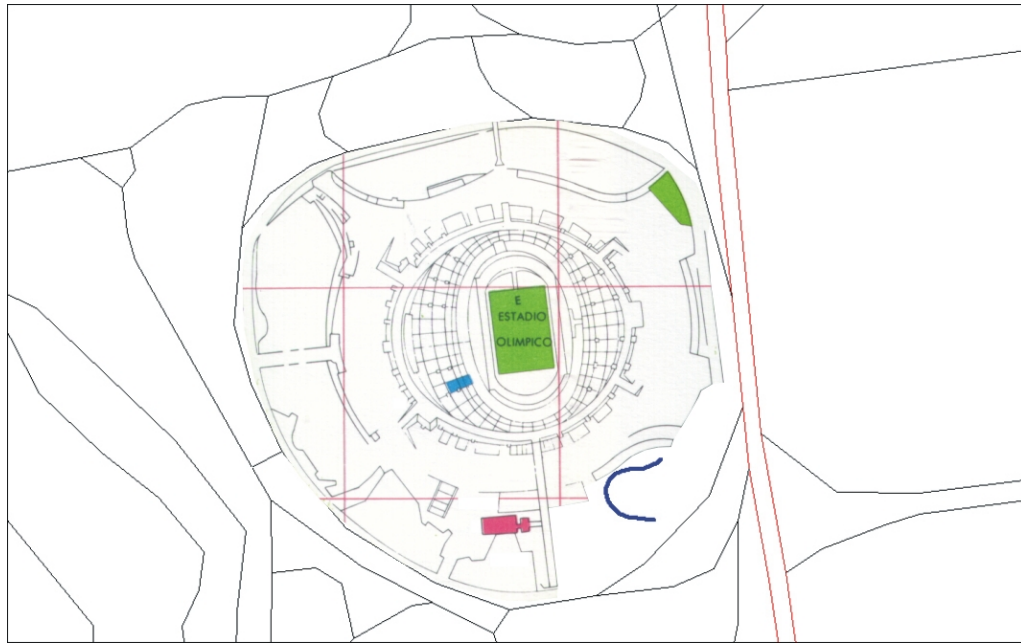


Figura 6.9 prueba de trayectoria 2 partiendo a 45° con respecto a la dirección contraria del punto destino en el estacionamiento 8 del estadio olímpico

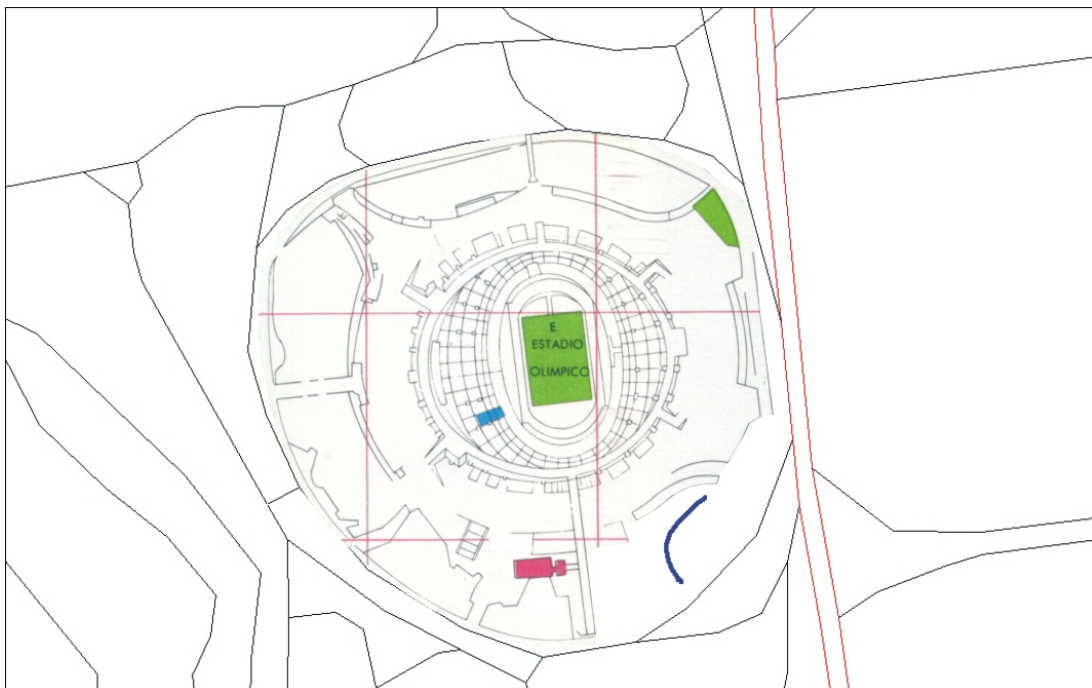


Figura 6.10 prueba de trayectoria 23 partiendo a 90° con respecto a la dirección contraria del punto destino en el estacionamiento 8 del estadio olímpico

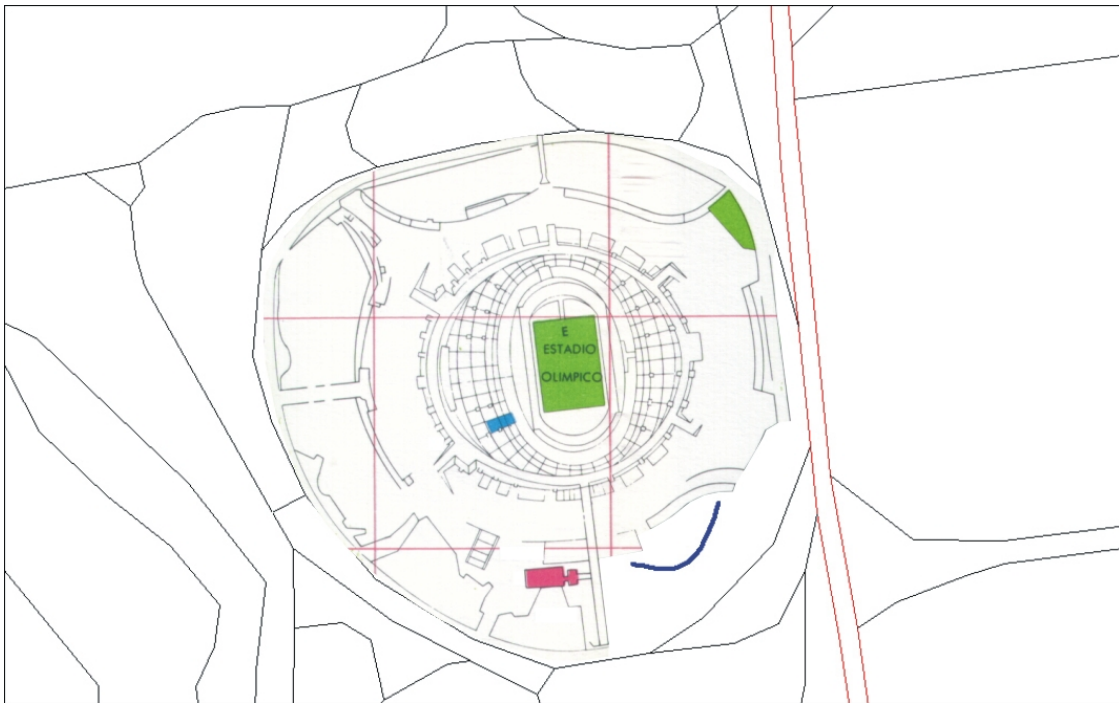


Figura 6.13 prueba de trayectoria 6 partiendo a 225° con respecto a la dirección contraria del punto destino en el estacionamiento 8 del estadio olímpico

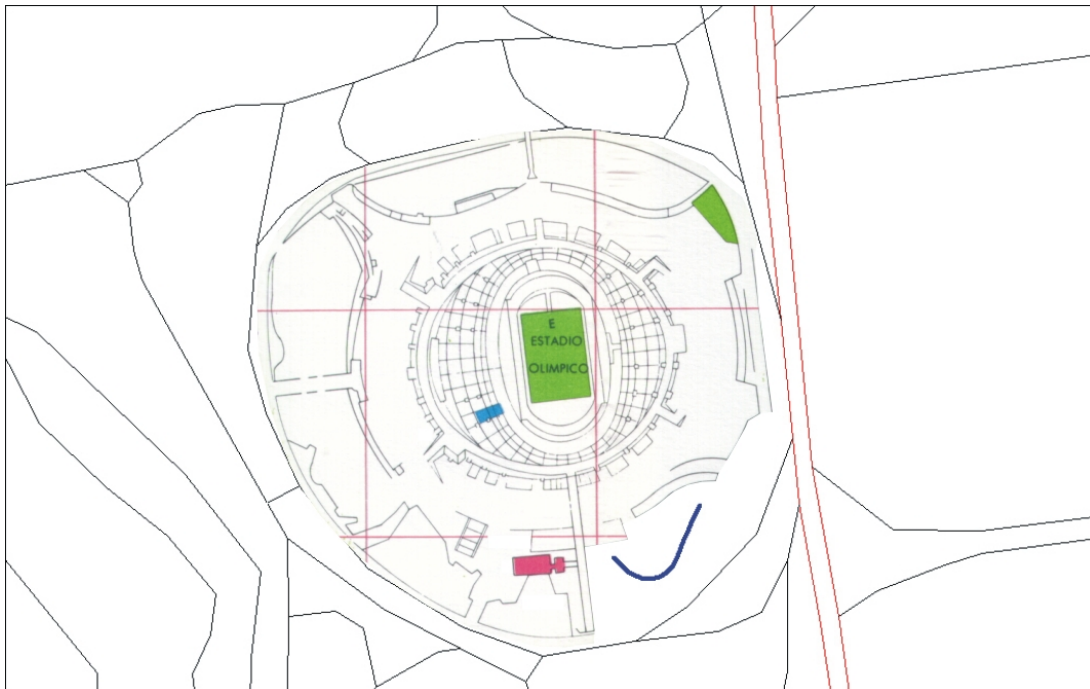


Figura 6.14 prueba de trayectoria 7 partiendo a 270° con respecto a la dirección contraria del punto destino en el estacionamiento 8 del estadio olímpico

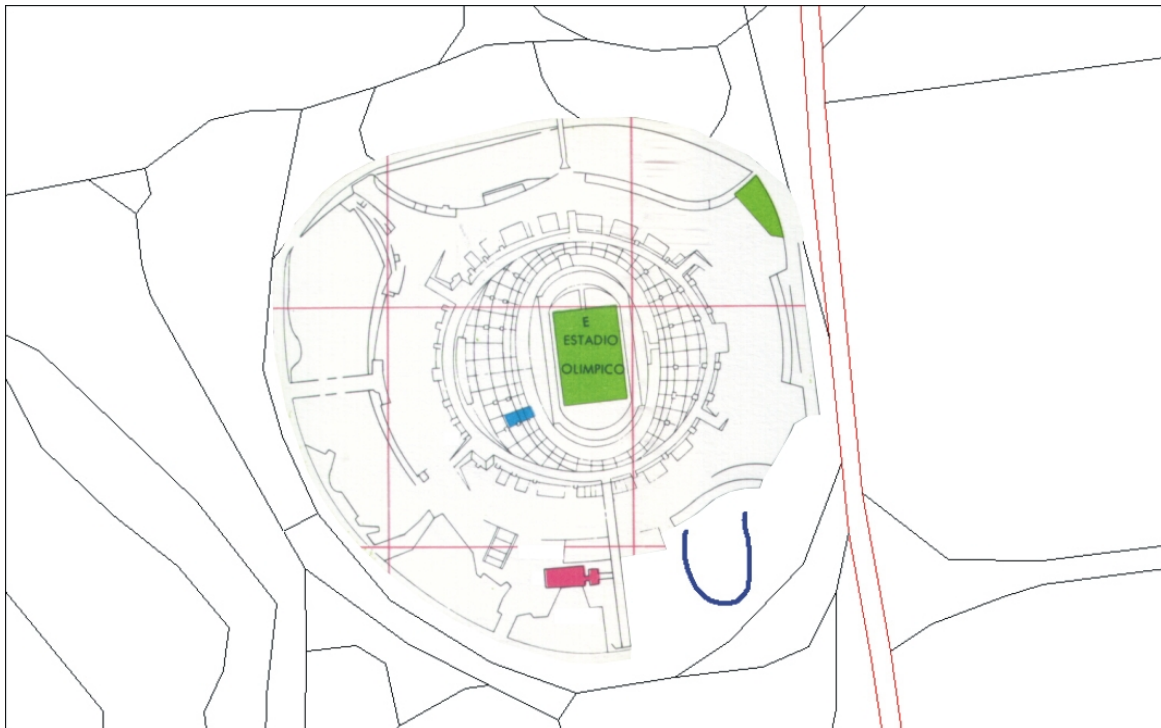


Figura 6.15 prueba de trayectoria 8 partiendo a 315° con respecto a la dirección contraria del punto destino en el estacionamiento 8 del estadio olímpico

Como resultado de esta prueba se obtuvieron diferentes códigos como salida del sistema, que indicaban las correcciones en la dirección necesarias para alcanzar el móvil, los códigos que se obtuvieron de estas pruebas son los siguientes:

Tabla 6.2

Código	Instrucción de corrección
X,1,0,1	Avanza y derecha
X,1,0,2	Avanza e Izquierda
X,1,0,0	Avanza recto
0,0,0,0	Paro

Estos códigos son la causa de las correcciones de dirección hechas en el vehículo.

6.2.4 Pruebas de Aproximación

Las pruebas de aproximación tienen como objetivo evaluar el comportamiento del sistema cuando este realiza la aproximación al punto de destino en línea recta desde diferentes direcciones. Para estas pruebas se realizaron trayectos hacia un mismo punto en el estacionamiento numero 8 del estadio olímpico México 68 de ciudad universitaria. Se colocó el punto N $19^{\circ} 19.803$ W $99^{\circ} 11.461$ con una altitud de 2290 m de altitud, cada una de las 8 pruebas tiene una diferencia en dirección de 45° desde el punto origen hacia el punto destino. En estas se observaron los códigos emitidos por el sistema como parte del calculo de corrección de dirección. Para esta prueba se esperó como resultado siempre el mismo código de dirección X,1,0,0 indicando que se avance en línea recta puesto que el punto final siempre se encontró al frente del vehículo en dirección recta.

Las observaciones GPS generadas por el receptor para realizar estas pruebas fueron registradas e interpretadas gráficamente en un mapa de ciudad universitaria. Las imágenes generadas se presentan a continuación en esta sección.

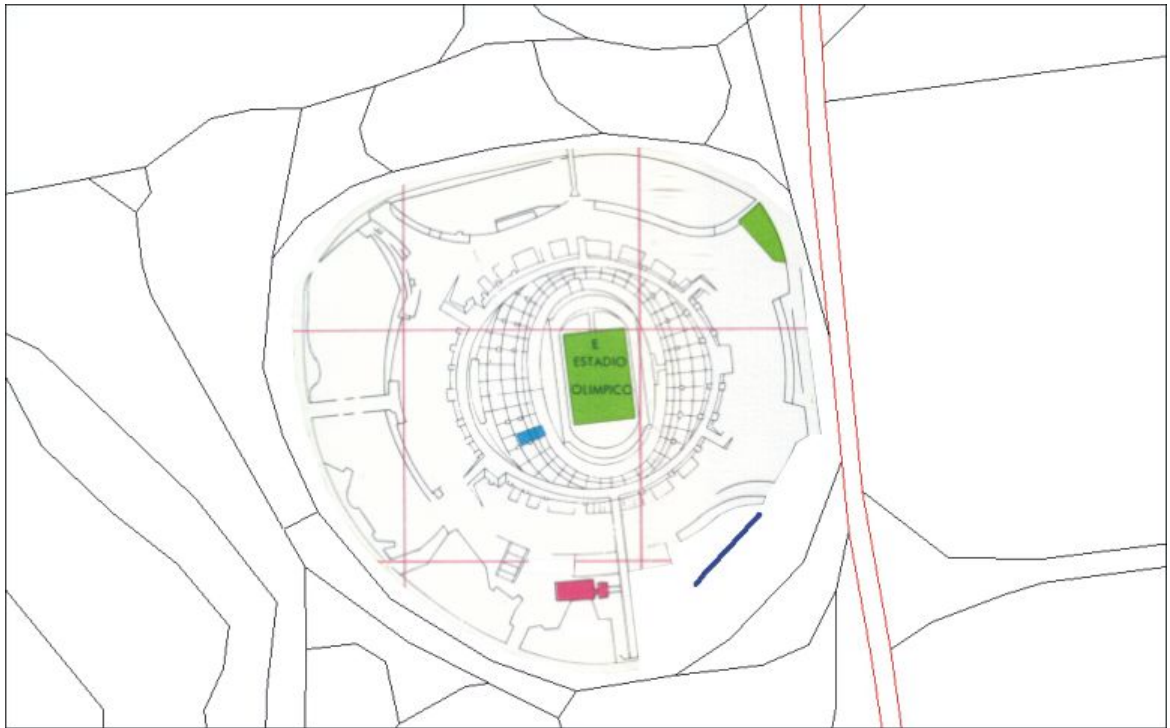


Figura 6.16 prueba de aproximación 1 en el estacionamiento 8 del estadio olímpico

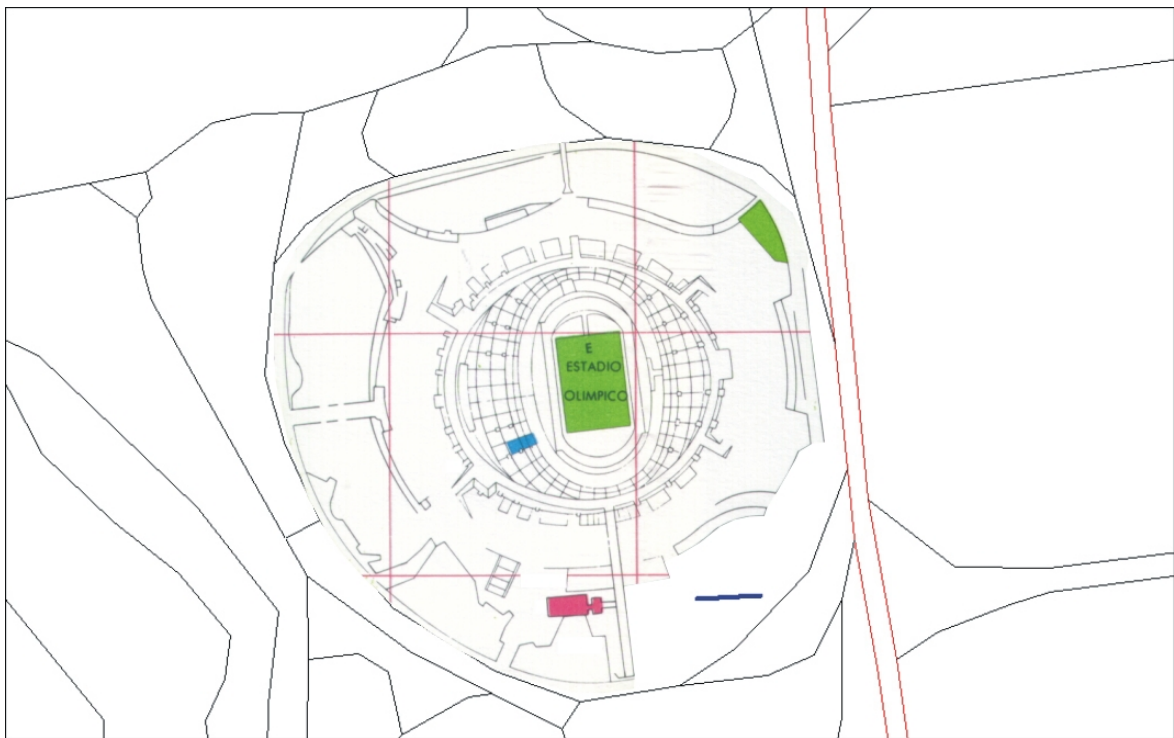


Figura 6.17 prueba de aproximación 2 en el estacionamiento 8 del estadio olímpico

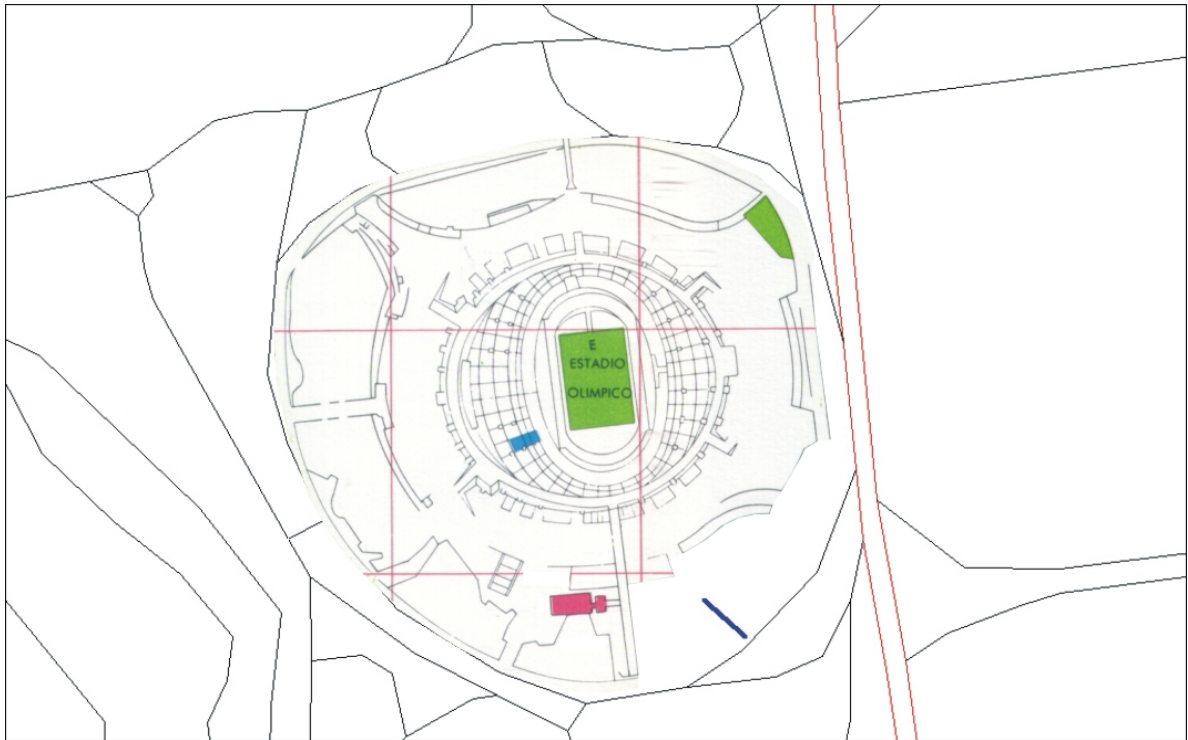


Figura 6.18 prueba de aproximación 3 en el estacionamiento 8 del estadio olímpico

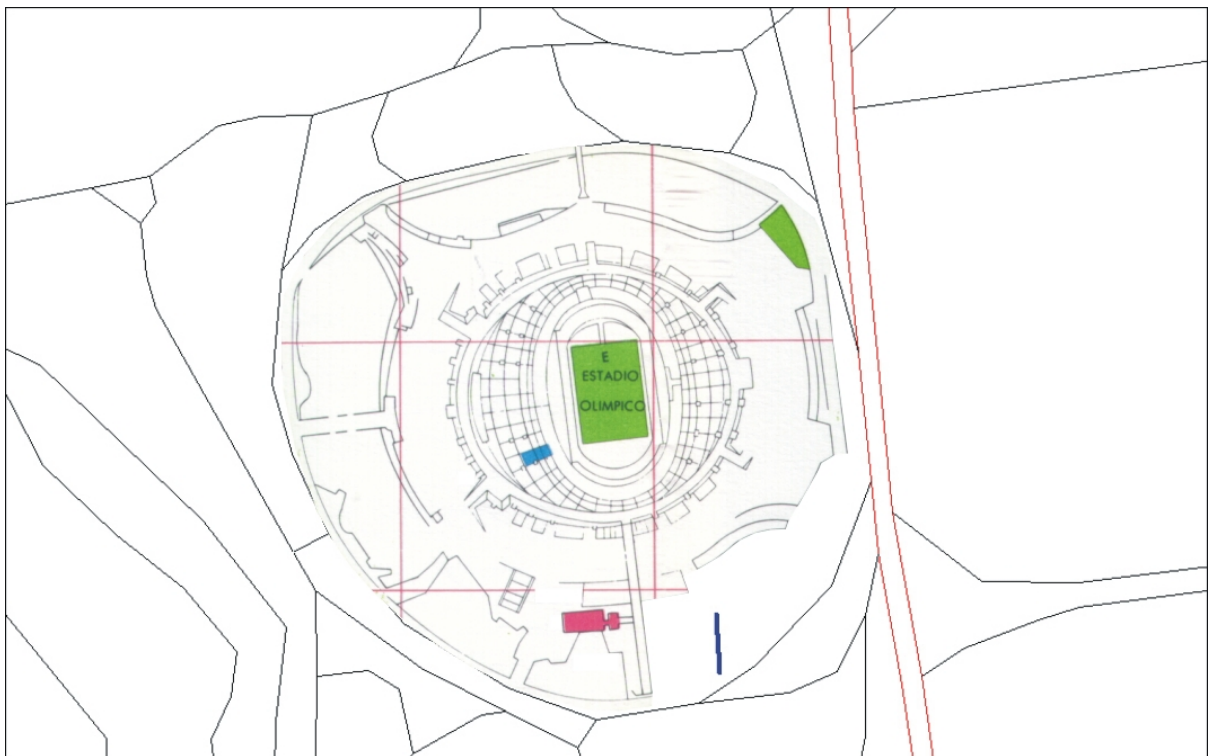


Figura 6.19 prueba de aproximación 4 en el estacionamiento 8 del estadio olímpico

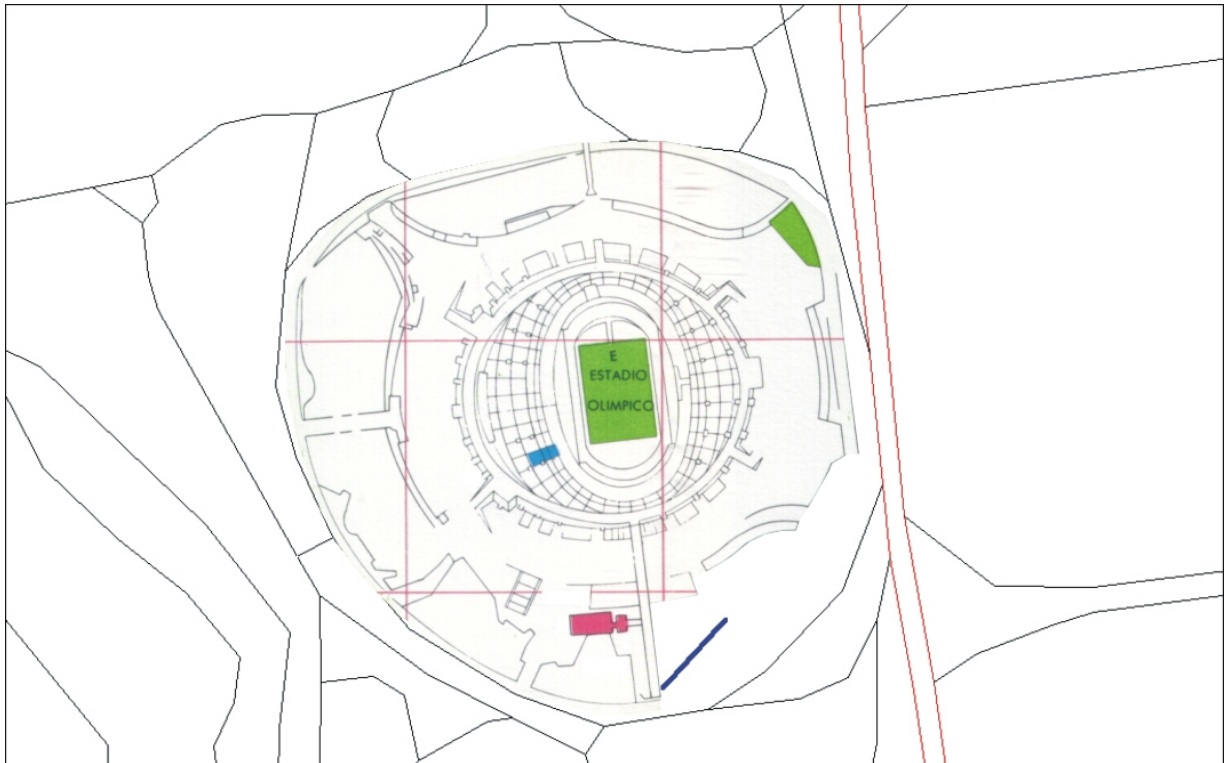


Figura 6.20 prueba de aproximación 5 en el estacionamiento 8 del estadio olímpico

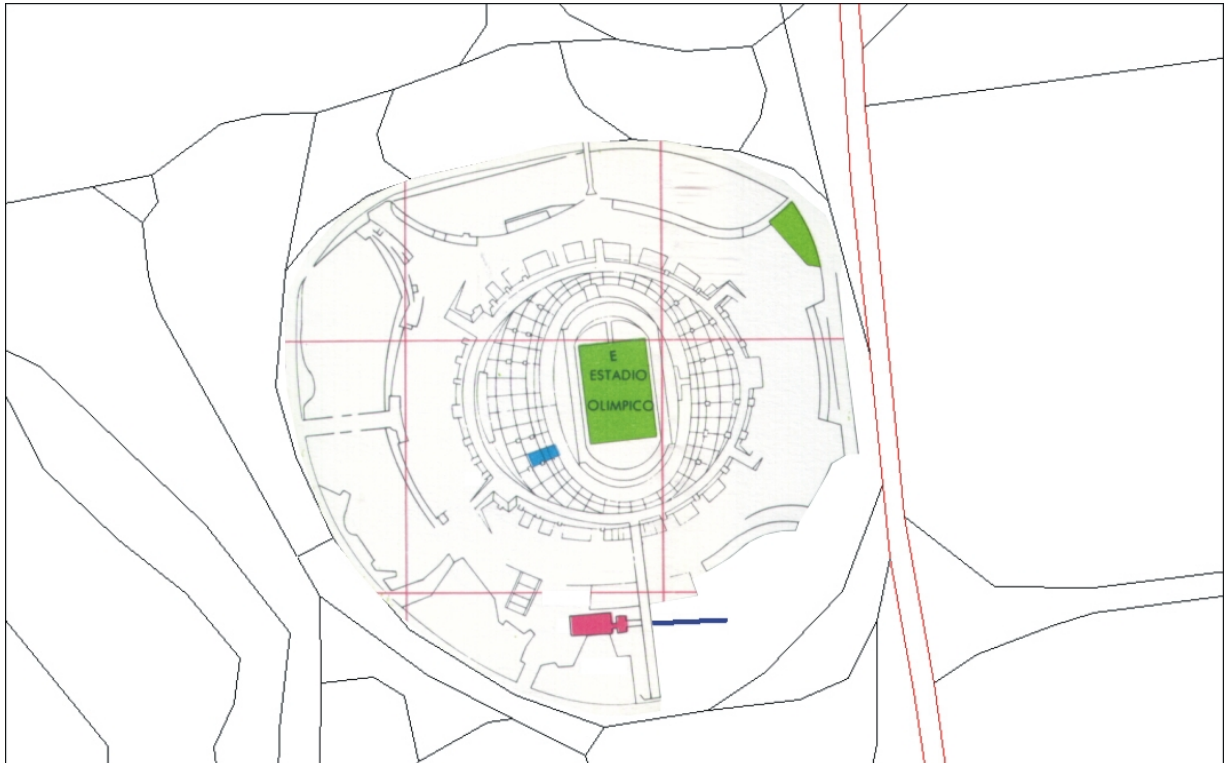


Figura 6.21 prueba de aproximación 6 en el estacionamiento 8 del estadio olímpico

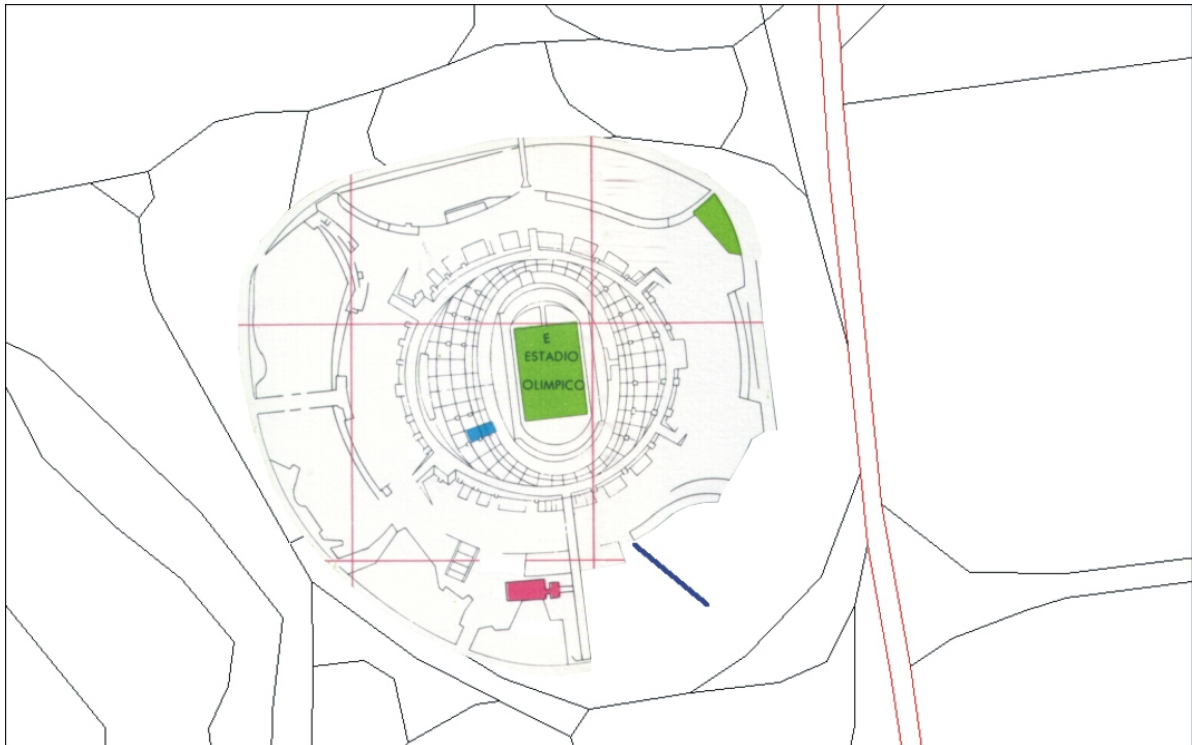


Figura 6.22 prueba de aproximación 7 en el estacionamiento 8 del estadio olímpico

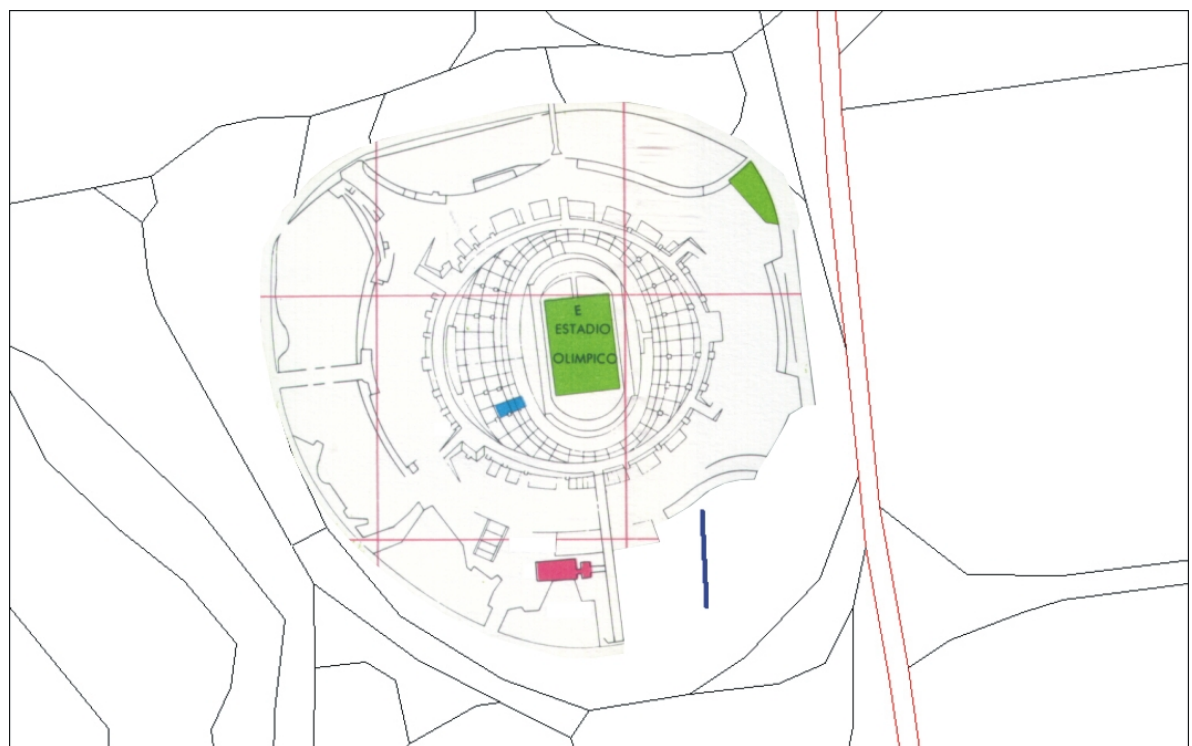


Figura 6.23 prueba de aproximación 8 en el estacionamiento 8 del estadio olímpico

En estas pruebas también se evaluó el límite de acercamiento del sistema hacia el punto destino. El límite del sistema está programado a 10 m del punto de destino y en todas las pruebas el sistema detuvo el vehículo a una distancia de entre 12 y 8 m del punto destino debido al error inherente del sistema GPS pero esta distancia es aceptable para los fines del sistema.

7. CONCLUSIONES 1

7. Conclusiones.

Haciendo un análisis de los resultados obtenidos a lo largo de las pruebas del sistema autómatas de navegación, se concluye que el desempeño del mismo es el esperado al realizar las correcciones de dirección necesarias a un móvil, para que este pueda alcanzar su destino, el cual fue previamente seleccionado partiendo desde un punto cualquiera

Evaluando los resultados obtenidos del sistema en pruebas desarrolladas en distancias pequeñas, podemos determinar que el sistema funciona perfectamente lo que le permitiría desplazarse a lo largo de distancias relativamente grandes, ya que realiza las correcciones de dirección con una gran precisión, .

Debido al desempeño y al comportamiento del sistema durante el transcurso de las pruebas realizadas, podemos concluir, que su desempeño bajo las condiciones imperantes en México es muy bueno tomando en cuenta el error inherente del sistema GPS en nuestro país.

Así mismo, este sistema puede ser adaptado o incorporado a cualquier tipo de vehículo ya sea que se desplace en dos o tres dimensiones, por lo que fácilmente puede incorporarse para controlar el movimiento de un vehículo terrestre manipulando el o los motores de dirección y de propulsión o desplazamiento. También puede incorporarse de una manera relativamente sencilla a un vehículo acuático, controlando el o los motores de propulsión,

los mismos que controlarían la dirección. Y por último, a un vehículo aéreo, controlando los motores de propulsión, dirección y elevación, en este caso se recomienda la instalación por su sencillez, a un dirigible de helio. Hay que hacer notar que el control de los motores debe realizarse a partir de la conexión de motores eléctricos o servos a la etapa de potencia del sistema.

Partiendo de la conclusión de que el sistema puede ser implementado en un vehículo ya sea de características acuáticas, terrestres o aéreas, que se desplace en espacios abiertos, las aplicaciones que se le pueden dar al mismo son muchas, por ejemplo, en un vehículo terrestre se puede incorporar el sistema para que realice viajes autónomos a lugares donde no sería posible enviar un humano, ejemplo de esto sería, a lugares donde han ocurrido desastres bacteriológicos o peor aun, accidentes nucleares, donde, por los altos niveles de radioactividad en el ambiente; no es posible enviar unidades con humanos, se puede enviar el vehículo de manera autónoma a recoger tanto muestras como imágenes, de igual forma el vehículo puede ser programado para que una vez que el mismo ha alcanzado su destino inicie el viaje de regreso al punto de origen.

También pudiera implementarse un vehículo acuático de pequeñas dimensiones, el cual puede ser programado para recolectar imágenes en el caso de un derrame petrolero o un accidente en una plataforma petrolera, como parte del reconocimiento antes de enviar barcos mas grandes con personas. Otro ejemplo pudiera ser, programarlo para recorrer de manera ininterrumpida una aérea determinada en busca de sobrevivientes de algún naufragio, y así,

hacer la búsqueda mas efectiva al combinar varios de estos vehículos autónomos, con los navíos convencionales de las guardias costeras que realizan las búsquedas hoy en día.

Al igual que en las aplicaciones anteriores se puede construir un vehículo aéreo autónomo equipado con cámaras y sensores de gases y ser programado para sobrevolar áreas y tomar muestras e imágenes de lugares, donde seria muy peligros el envío de helicópteros tripulados, como los son: los cráteres de volcanes activos, zonas donde han ocurrido desastres bacteriológicos o accidentes nucleares, así mismo puede ser muy útil para el patrullaje de zonas especificas ya que puede programarse para ser enviado a un cierto punto y una vez que lo alcance regresar al punto inicial, y realizar este recorrido una y otra vez de manera ininterrumpida, este vehículo puede ser equipado con equipos de video y radares para monitorear todo tipo de vehículos que transiten por esa zona especifica.

Como se ha venido observando a lo largo de este trabajo de investigación podemos concluir que los objetivos planteados al inicio del mismo, se han cumplido satisfactoriamente, esperando que el desarrollo de este trabajo ayude al entendimiento de los sistemas GPS y de algunas de las aplicaciones que se pueden desarrollar con base en los mismos sistemas.

APÉNDICE 1

PROGRAMA ASM. 1

Apéndice .

Programa ASM.

```

*PROGRAMA DE CONTROL DEL SISTEMA AUTOMATA DE NAVEGACION BASADO EN EL SISTEMA DE POSICIONAMIENTO
*GLOBAL GPS

*PROGRAMA QUE CARGA LOS DATOS FINALES IDENTIFICA LA CADENA GPGGA Y TRANSFORMA LOS DATOS A
*HEXADECIMAL DE 16 BITS
*REALIZA LOS DELTAS Y CALCULA LA TANGENTE Y REALIZA EL CONTROL DEL SISTEMA

*COMIENZA EN LA $1100 PONE 00 EN LOS MOTORES CUANDO SE ACERCA DERECHO AL DESTINO

*Coneccion: se conecta el puerto A al driver que controla los motores de direccion
*
*           y se controla con $01 para derecha y $02 para izquierda
*
*           se conecta el puerto G al driver que controla los motores de altitud y
*
*           principal se controlan con $10 para subir $20 para bajar y $01 para avanzar

*SI EL SIGNO ES S=$00 O +
*SI EL SIGNO ES S=$01 O -

*CONFIGURACIONES *
BAUD   EQU $2B           ;configura la velocidad de trnsmision
SCCR1  EQU $2C           ;configura el protocolo y tama?o del dato
SCCR2  EQU $2D           ;habilita transmision/recepcion de e interrupciones
SCSR   EQU $2E           ;registro de estado, contiene las banderas de transmision y recepcion
                           ;completa
SCDR   EQU $2F           ;registro de datos transmision/recepcion

*VALORES *
POSI   EQU $3A2          ;posicion inicial donde comienza a guardar la trama GPGGA $3A2
POSF   EQU $3E8          ;posicion final donde guarda la trama GPGGA

**POSICIONES FINALES**

LATF   EQU   POSI-42     ;latitud instantanea HEX 16-bits           $360
LONF   EQU   POSI-40     ;longitud instantanea HEX 16-bits          $362
ALTF   EQU   POSI-3E     ;altitud instantanea HEX 16-bits          $364

*VARIABLES

FLAG   EQU   POSI-6      ;bandera para identificacion de trama GPGGA           $39C
FLAG1  EQU   POSI-5      ;bandera                                               $39D
INDICE EQU   POSI-4      ;variable que guarda el indice X mientras la interrupcion $39E
INDSEPX1 EQU POSI-12     ;variable que guarda el indice X mientras actualiza los datos $390
FLAG2  EQU   POSI-10     ;bandera                                               $392

*POSICIONES *

*LATIN EQU $3AF          ;posicion donde comienzan los datos de latitud en la trama GPGGA
*LATIN1 EQU $360         ;posicion donde comienzan los datos instantaneos de latitud
*ALTIN EQU $3D2          ;posicion donde comienzan los datos de altitud en la trama GPGGA
*ALTIN1 EQU $359        ;posicion donde comienzan los datos instantaneos de altitud

*SIGNOS
SDELLR EQU   POSI+4E     ;signo del delta de latitud de rumbo latf-latins      $3F0
SDELLnR EQU  POSI+4F     ;signo del delta de longitud de rumbo longf-longins    $3F1

```

```

SDELLD EQU     POSI+50     ;signo del delta de latitud de direccion latf-latins    $3F2
SDELLnD EQU    POSI+51     ;signo del delta de longitud de direccion longf-longins    $3F3
SDIFTAN EQU    POSI+52     ;signo de la diferencia de altitud                          $3F4

*DELTAS
DELLR EQU     POSI-2A     ;delta de latitud de rumbo latf-latins 16-bits              $378
DELLnR EQU    POSI-28     ;delta de longitud de rumbo longf-longins 16-bits           $37A
DELLD EQU     POSI-1A     ;delta de latitud de direccion latins-latant 16-bits        $388
DELLnD EQU    POSI-18     ;delta de longitud de direccion longins-longant 16-bits     $38A
DIFALT EQU    POSI-14     ;diferencia de altitud                                      $38E
DIFTAN EQU    POSI-25     ;diferencia de tangentes                                    $37D

*TANGENTES
TANR EQU     POSI-36     ;                                                              $36C
TAND EQU     POSI-34     ;                                                              $36E

*POSICIONES INSTANTANEAS
LATINS EQU    POSI-32     ;latitud instantanea HEX 16-bits                            $370
LONINS EQU    POSI-30     ;longitud instantanea HEX 16-bits                           $372
ALTINS EQU    POSI-2E     ;altitud instantanea HEX 16-bits                            $374

*POSICIONES ANTERIORES
LATANT EQU    POSI-22     ;latitud anterior HEX 16-bits                               $380
LONANT EQU    POSI-20     ;longitud anterior HEX 16-bits                              $382
ALTANT EQU    POSI-1E     ;altitud anterior HEX 16-bits                               $384

*TEMPORALES
TEMP1 EQU     POSI+58     ;variable temporal                                          $3FA
TEMP2 EQU     POSI+5A     ;variable temporal                                          $3FC
TEMP3 EQU     POSI+5C     ;variable temporal                                          $3FE

T1 EQU     POSI+53     ;temporal para transformar el dato de 5 cifras              $3F5
T2 EQU     POSI+54     ;una cifra por posicion                                     $3F6
T3 EQU     POSI+55     ;                                                            $3F7
T4 EQU     POSI+56     ;                                                            $3F8
T5 EQU     POSI+57     ;                                                            $3F9

*POSICIONES DE LOS DATOS EN LA CADENA GPGGA ORIGINAL
LA1 EQU     POSI+0F     ;minutos y sus decimales de la latitud de 5 cifras         $3B1
LA2 EQU     POSI+10     ;una cifra por posicion                                     $3B2
LA3 EQU     POSI+12     ;                                                            $3B4
LA4 EQU     POSI+13     ;                                                            $3B5
LA5 EQU     POSI+14     ;                                                            $3B6

LN1 EQU     POSI+1C     ;minutos y sus decimales de la longitud de 5 cifras        $3BE
LN2 EQU     POSI+1D     ;una cifra por posicion                                     $3BF
LN3 EQU     POSI+1F     ;                                                            $3C1
LN4 EQU     POSI+20     ;                                                            $3C2
LN5 EQU     POSI+21     ;                                                            $3C3

AL1 EQU     POSI+30     ;metros de altitud de 4 cifras                              $3D2
AL2 EQU     POSI+31     ;una cifra por posicion                                     $3D3
AL3 EQU     POSI+32     ;                                                            $3D4
AL4 EQU     POSI+33     ;                                                            $3D5

    ORG     $1100
    LDS     #$0FFF         ;inicializa la pila
    LDY     #$1000        ;inicializa el indice en 1000
    CLR     FLAG1

*CONFIGURA PSEUDOVECTOR DE INTERRUPCION PARA LA CARGA DE DATOS FINALES*
INICIO LDAA   #$7E         ;direccion del pseudovector de int.
        STAA   $C4         ;guarda en inicio del pseudovector
        LDX   #SET_DATA    ;guarda direccion de subrutina
        STX   $C5         ;en el pseudovector

* CONFIGURA PUERTO A *
    LDAA   #$FF           ;configura el puerto A de salida
    STAA   $01,Y         ;configura el puerto G de salida
    STAA   $03,Y         ;configura el puerto D de salida
    STAA   $09,Y

* CONFIGURA SCI *

```



```

LDAA    #$31          ;configura a 4800 bauds
STAA    BAUD,Y
LDAA    #$00          ;configura lbit inicio i bit stop
STAA    SCCR1,Y      ;8 bits de datos
LDAA    #$2C          ;habilita int. por recepcion, TE/RE
STAA    SCCR2,Y

* INICIO DEL PROGRAMA
LDAA    #$00          ;inicia la bandera de inicio de carga de datos
STAA    $FF,Y
LDAB    #$00
STAB    FLAG          ; inicializa las banderas
STAB    FLAG2
LDD     #$FFFF        ;limpia el acumulador D
STD     DIFALT        ;la diferencia de altura
STD     DELLR         ;limpia el delta de latitud de Rumbo
STD     DELLnR        ;limpia el delta de longitud de Rumbo
LDAB    #$24          ;pone un signo de $ en el inicio de la sentencia GPGGA
STAB    POSI-1
LDX     #T1           ;carga la posicion de donde va a
STX     INDICE        ;guardar los datos de la cadena GPGGA
BCLR    SCSR,Y,$20    ;limpia la bandera de recepcion completa
CLI     ;habilita interrupciones generales
LDAB    #$00          ;pone en 0 todos los puertos de salida
STAB    $00,Y
STAB    $02,Y
ESPERA BRCLR    $FF,Y,$01,ESPERA ;espera a que la bandera de inicio sea 1

*PROGRAMA DE DIRECCION
MAIN    JSR     AVANZA ;envia a la rutina que hace avanzar el vehiculo
LDD     DIFALT        ;si la diferencia de altitud es mayor a 50 metros
CPD     #$32          ;pasa al control de altitud
BHI     SIG_ALL1     ;si no lo es continua el programa
LDD     DELLR         ;verifica si el delta de latitud de
CPD     #$0A          ;Rumbo es menor a 10 metros
BLS     SIG_PAR      ;verifica la longitud
SIG_1   LDD     DELLR        ;si delta de latitud de Rumbo es 0
CPD     #$00
BEQ     SIG_01        ;envia a los casos de 0
LDD     DELLnR        ;si delta de longitud de Rumbo es 0
CPD     #$00
BEQ     SIG_03_2      ;envia a los casos de 0
LDD     DELLD         ;si delta de latitud de Direccion es 0
CPD     #$00          ;pasa al caso G
BEQ     G1
LDD     DELLnD        ;si delta de longitud de Direccion es 0
CPD     #$00          ;pasa al caso H
BEQ     H1
BNE     SIG_2_2        ;de no ser asi el programa continua

SIG_PAR LDD     DELLnR        ;verifica si el delta de longitud de Rumbo
CPD     #$0A          ;es menor a 10 metros
BLS     PARO          ;va a la rutina de paro
JMP     SIG_1

PARO    LDAA    #$00          ;Envia el codigo al motor de principal
STAA    $00,Y          ;para que se apague
STAA    $02,Y          ;para que apague los motores
LDAB    #$01          ;pone la bandera de inicio en 1
STAB    FLAG1
JSR     SAL_LIN        ;Salta a salto de linea
JSR     SAL_LIN        ;Salta a salto de linea
JSR     FINAL         ;escribe la llegada al punto destino
JSR     SAL_LIN        ;Salta a salto de linea
JMP     INICIO        ;finaliza el programa y vuelve a empezar

```

* PUENTES PARA ALCANZAR LAS RUTINAS

```

E1      JMP      E          ;puentea para que alcance la rutina E
F1      JMP      F          ;puentea para que alcance la rutina F
G1      JMP      G          ;puentea para que alcance la rutina G
H1      JMP      H          ;puentea para que alcance la rutina H
DER1    JMP      DER        ;puentea para que alcance la rutina DER
IZQ1    JMP      IZQ        ;puentea para que alcance la rutina IZQ
MAIN1   JMP      MAIN       ;puentea para que alcance la rutina MAIN
SIG_AL1 JMP      SIG_AL2     ;puentea para que alcance la rutina SIG_AL
SIG_2_2 JMP      SIG_2       ;puentea para que alcance la rutina SIG_2
SIG_03_2 JMP     SIG_3       ;puentea para que alcance la rutina SIG_3

```

*CUANDO ALGUN DELTA ES CERO

```

SIG_01  LDD      DELLD      ;si delta de latitud de Direccion es 0
        CPD      #$00       ;compara los signos de delta LD y LnR
        BNE      SIG_02     ;si no es 0 compara delta LnD
        LDAA     SDELLnR
        CMPA     SDELLnD     ;si el signo de delta LnD es igual al de LnR
        BEQ      MAIN1      ;si son iguales va al prog principal casos I1,I3
        BNE      DER1       ;si son desiguales va a DER casos I2,I4

SIG_02  LDD      DELLnD     ;si delta de longitud de Direccion es 0
        CPD      #$00       ;compara los signos de delta LD y LnR
        BNE      E1         ;va al caso E
        LDAA     SDELLnR
        CMPA     SDELLD     ;si el signo de delta LD es igual al de LnR
        BEQ      IZQ1      ;si son iguales va a IZQ casos J1,J3
        BNE      DER1       ;si son desiguales va a DER casos J2,J4

SIG_03  LDD      DELLnD     ;si delta de longitud de Direccion es 0
        CPD      #$00       ;compara los signos de delta LD y LnR
        BNE      SIG_04     ;si no es 0 compara delta LnD
        LDAA     SDELLR
        CMPA     SDELLD     ;si el signo de delta LD es igual al de LR
        BEQ      MAIN1      ;si son iguales va al prog principal casos K1,K3
        BNE      DER1       ;si son desiguales va a DER casos K2,K4

SIG_04  LDD      DELLD      ;si delta de latitud de Direccion es 0
        CPD      #$00       ;compara los signos de delta LR y LnD
        BNE      F1         ;va al caso E
        LDAA     SDELLR
        CMPA     SDELLnD    ;si el signo de delta LR es igual al de LnD
        BEQ      DER1       ;si son iguales va a DER casos L1, L3
        BNE      IZQ1      ;si son desiguales va a IZQ casos L2, L4

```

*CUANDO LOS DELTAS NO SON CEROS

```

SIG_2   LDAA     SDELLR      ;checa el signo de delta de latitud de Rumbo
        CMPA     #$00        ;
        BNE      SIG_3       ;verifica el signo de delta longitud de Rumbo
        LDAA     SDELLnR     ;si delta de latitud de Rumbo LR es positivo
        CMPA     #$00        ;
        BEQ      B           ;y delta de longitud de Rumbo LnR es positivo pasa al caso B
        BNE      A           ;y delta de longitud de Rumbo LnR es negativo pasa al caso A

SIG_3   LDAA     SDELLnR     ;si delta de latitud de Rumbo LR es negativo
        CMPA     #$00        ;

```

```

      BEQ      C          ;y delta de longitud de Rumbo LnR es positivo pasa al caso C
      BNE      D          ;y delta de longitud de Rumbo LnR es negativo pasa al caso D

*CONTROL DE ALTITUD

SIG_AL2 LDAA      ALTF          ;si la altitud final es mayor que la inicial
      CMPA      ALTINS         ;sube
      BHI      SUBE
      BLO      BAJA          ;si la altitud final es mayor que la inicial baja
      BEQ      MAIN1         ;si el igual regresa al programa principal

*SUBE

SUBE     LDAB      #$10          ;Envia el codigo al motor de altitud
      STAB      $02,Y          ;para que gire hacia arriba
      STAB      $34E          ;graba la accion del motor
      JMP      MAIN          ;y va al programa principal

*BAJA

BAJA     LDAB      #$20          ;Envia el codigo al motor de altitud
      STAB      $02,Y          ;para que gire hacia abajo
      STAB      $34E          ;graba la accion del motor
      JMP      MAIN          ;y va al programa principal

*CASO A
A        LDAA      SDELLD          ;verifica el signo de delta Ld
      CMPA      #$00
      BNE      SIG_A1
      LDAA      SDELLnD          ;si delta LD es positivo
      CMPA      #$00
      BEQ      DER2          ;y delta LnD positivo va a DER, A2
      BNE      DES1_2          ;y delta LnD negativo va a DES1, A1

SIG_A1   LDAA      SDELLnD          ;si delta LD es negativo
      CMPA      #$00
      BEQ      DES22          ;y delta LnD positivo va a DES22, A3
      BNE      IZQ2          ;y delta LnD negativo va a IZQ, A4

*CASO B
B        LDAA      SDELLD          ;verifica el signo de delta Ld
      CMPA      #$00
      BNE      SIG_B1
      LDAA      SDELLnD          ;si delta LD es positivo
      CMPA      #$00
      BEQ      DES2_2          ;y delta LnD positivo va a DES2, B2
      BNE      IZQ2          ;y delta LnD negativo va a IZQ, B1

SIG_B1   LDAA      SDELLnD          ;si delta LD es negativo
      CMPA      #$00
      BEQ      DER2          ;y delta LnD positivo va a DER, B3
      BNE      DES12          ;y delta LnD negativo va a DES12, B4

*CASO C
C        LDAA      SDELLD          ;verifica el signo de delta Ld
      CMPA      #$00
      BNE      SIG_C1
      LDAA      SDELLnD          ;si delta LD es positivo
      CMPA      #$00
      BEQ      IZQ2          ;y delta LnD positivo va a IZQ, C2
      BNE      DES22          ;y delta LnD negativo va a DES22, C1

SIG_C1   LDAA      SDELLnD          ;si delta LD es negativo
      CMPA      #$00
      BEQ      DES1_2          ;y delta LnD positivo va a DES1, C3

```

```

BNE     DER2             ;y delta LnD negativo va a DER,  C4

*CASO D
D       LDAA     SDELLD     ;verifica el signo de delta Ld
        CMPA     #$00
        BNE     SIG_D1
        LDAA     SDELLnD     ;si delta LD es positivo
        CMPA     #$00
        BEQ     DES122     ;y delta LnD positivo va a DES12, D2
        BNE     DER2       ;y delta LnD negativo va a DER,  D1

SIG_D1  LDAA     SDELLnD     ;si delta LD es negativo
        CMPA     #$00
        BEQ     IZQ2       ;y delta LnD positivo va a IZQ,  D3
        BNE     DES2_2     ;y delta LnD negativo va a DES2, D4

*PUENTES PARA QUE EL PROGRAMA ALCANCE LAS RUTINAS

DER2    JMP     DER         ;puentea para que alcance la rutina DER
IZQ2    JMP     IZQ        ;puentea para que alcance la rutina IZQ
MAIN2   JMP     MAIN       ;puentea para que alcance la rutina MAIN
DES1_2  JMP     DES1       ;puentea para que alcance la rutina DES1
DES2_2  JMP     DES2       ;puentea para que alcance la rutina DES2
DES122  JMP     DES12      ;puentea para que alcance la rutina DES12
DES222  JMP     DES22      ;puentea para que alcance la rutina DES22

*CASO E
E       LDAA     SDELLnR     ;verifica el signo de delta LnR
        CMPA     #$00
        BNE     SIG_E1
        LDAA     SDELLD     ;si delta LnR es positivo
        CMPA     #$00
        BEQ     IZQ3       ;y delta LD positivo va a IZQ,  E3
        BNE     DER3       ;y delta LD negativo va a DER,  E4

SIG_E1  LDAA     SDELLD     ;si delta LnR es negativo
        CMPA     #$00
        BEQ     DER3       ;y delta LD positivo va a DER,  E1
        BNE     IZQ3       ;y delta LD negativo va a IZQ,  E2

IZQ3    JMP     IZQ        ;puentea para que alcance la rutina IZQ
DER3    JMP     DER        ;puentea para que alcance la rutina DER

*CASO F
F       LDAA     SDELLR     ;verifica el signo de delta LR
        CMPA     #$00
        BNE     SIG_F1
        LDAA     SDELLnD     ;si delta LR es positivo
        CMPA     #$00
        BEQ     DER        ;y delta LnD positivo va a DER,  F4
        BNE     IZQ        ;y delta LnD negativo va a IZQ,  F3

SIG_F1  LDAA     SDELLnD     ;si delta LR es negativo
        CMPA     #$00
        BEQ     IZQ        ;y delta LnD positivo va a IZQ,  F2
        BNE     DER        ;y delta LnD negativo va a DER,  F1

*CASO G
G       LDAA     SDELLnD     ;verifica el signo de delta LnD
        CMPA     #$00
        BNE     SIG_G1
        LDAA     SDELLR     ;si delta LnD es positivo
        CMPA     #$00
        BEQ     DER        ;y delta LR positivo va a DER,  G3

```

```

        BNE      IZQ          ;y delta LR negativo va a IZQ,  G4
SIG_G1  LDAA    SDELLR      ;si delta LnD es negativo
        CMPA    #$00
        BEQ     IZQ          ;y delta LR positivo va a IZQ,  G1
        BNE     DER          ;y delta LR negativo va a DER,  G2

*CASO H
H        LDAA    SDELLD      ;verifica el signo de delta LD
        CMPA    #$00
        BNE     SIG_H1
        LDAA    SDELLnR      ;si delta LD es positivo
        CMPA    #$00
        BEQ     IZQ          ;y delta LD positivo va a IZQ,  H4
        BNE     DER          ;y delta LD negativo va a DER,  H3

SIG_H1  LDAA    SDELLnR      ;si delta LnR es negativo
        CMPA    #$00
        BEQ     DER          ;y delta LD positivo va a DER,  H2
        BNE     IZQ          ;y delta LD negativo va a IZQ,  H1

* CONTROLES DE DIRECCION

DES1    LDAA    #$01          ;guarda en la memoria la decision que tomo
        STAA    $341
        LDD     DIFTAN
        CPD     #$03E8        ;si la diferencia de tangentes es
        BLS     NIVELA        ;menor a 1000 avanza recto
DES12   LDD     TANR          ;si las tangentes son iguales val al inicio de prog
        CPD     TAND          ;si la tangente de Rumbo es mayor que la
        BHI     IZQ          ;tangente de Direccion gira a la izquierda
        BLO     DER          ;si la tangente de Rumbo es menor que la
        BEQ     NIVELA        ;tangente de Direccion gira a la derecha
        ;y si es igual avanza

DES2    LDAA    #$02          ;guarda en la memoria la decision que tomo
        STAA    $341
        LDD     DIFTAN
        CPD     #$03E8        ;si la diferencia de tangentes es
        BLS     NIVELA        ;menor a 10000 avanza recto
DES22   LDD     TANR          ;si las tangentes son iguales val al inicio de prog
        CPD     TAND          ;si la tangente de Rumbo es mayor que la
        BHI     DER          ;tangente de Direccion gira a la derecha
        BLO     IZQ          ;si la tangente de Rumbo es menor que la
        BEQ     NIVELA        ;tangente de Direccion gira a la izquierda
        ;y si es igual avanza

DER     LDAA    #$01          ;Envia el codigo al motor de direccion
        STAA    $00,Y        ;para que gire a la derecha
        STAA    $34F        ;guarda en la memoria hacia donde debe girar
        JMP     MAIN        ;y va al programa principal

IZQ     LDAA    #$02          ;Envia el codigo al motor de direccion
        STAA    $00,Y        ;para que gire a la izquierda
        STAA    $34F        ;guarda en la memoria hacia donde debe girar
        JMP     MAIN        ;y va al programa principal

AVANZA  LDAA    #$01          ;Envia el codigo al motor de principal
        STAA    $02,Y        ;para que avance
        STAA    $34E        ;guarda en la memoria hacia donde se mueve el motor
        RTS                ;y regresa al programa

NIVELA  LDAA    #$00          ;Envia el codigo al motor de direccion
        STAA    $00,Y        ;para que no gire
        STAA    $34F        ;guarda en la memoria hacia donde debe girar
        JMP     MAIN        ;y va al programa principal

```

```

MAIN3   JMP     MAIN           ;pueneta para MAIN

*INTERRUPCION QUE GRABA LOS DATOS FINALES*

SET_DATA SEI                   ;deshabilita int generales
        BCLR   SCSR,Y,$20     ;limpia la bandera de recepcion completa
        LDAA  SCDR,Y         ;carga el dato recibido
        LDX   INDICE         ;carga el indice X
        LDAB  FLAG          ;carga el valor de la bandera contador
        CMPA  #' '           ;si el dato entrante es " " cuando de presiona
        BEQ   BIENES        ;la barra de espacio se escoge la bienvenida
        LDAA  FLAG2         ;compara la bandera de bienvenida para saber si
        CMPA  #$00          ;ya dio la bienvenida
        BNE   CARGA         ;y para a la carga de datos
        JMP   SALESET       ;sale de la interrupcion

*RUTINA QUE ESCOGE LA BIENVENIDA

BIENES  CLR    SCDR,Y        ;borra el registro del puerto serie
        LDAB  FLAG1         ;carga la bandera del tipo de bienvenida
        CMPB  #$00          ;cuando es 0 quiere decir que es la primera vez
        BEQ   BIEN1        ; que se ingresa al sistema si no ya se habia
        BNE   BIEN2        ;ingresado anteriormente

*RUTINA QUE ESCRIBE LA BIENVENIDA

BIEN1   LDAA  #$01          ;pone en 1 la bandera de bienvenida para
        STAA  FLAG2         ;indicar que ya se dio la bienvenida
        JSR   BIENV        ;va a la rutina que escribe la bienvenida
        JSR   CODEST       ;va a la rutina que escribe la carga de datos
        JSR   COOR         ;va a la rutina que escribe coordenadas del punto destino
        JMP   SALESET       ;sale de la interrupcion

*RUTINA QUE ESCRIBE LA NUEVA BIENVENIDA

BIEN2   LDAA  #$01          ;pone en 1 la bandera de bienvenida para
        STAA  FLAG2         ;indicar que ya se dio la bienvenida
        JSR   CODEST       ;va a la rutina que escribe la carga de datos
        JSR   NUEV         ;va a la rutina que escribe nuevo
        JSR   COOR         ;va a la rutina que escribe coordenadas del punto destino
        JMP   SALESET       ;sale de la interrupcion

*RUTINA QUE CARGA LOS DATOS FINALES

CARGA   LDAA  SCDR,Y        ;carga el dato recibido
        JSR   SERIAL       ;envia a la PC
        CMPA  #$6C         ;si el dato entrante es "l" cambia la
        BEQ   SALED1       ;interrupcion para recibir la trama GPGGA
        CMPA  #'r'         ;si el dato entrante es "r" recarga
        BEQ   RECAR        ;los datos finales
        CMPA  #$29         ;si el dato entrante no es un numero
        BLO  ERROR1        ;va a error y recarga
        CMPA  #$39         ;
        BHI  ERROR1        ;
        SUBA  #$30         ;transforma de ASCII a Hexa
        STAA  $00,X        ;guarda el dato en los temporales de datos de una cifra
        INCB  INCB         ;incrementa la bandera contadora
        INX   INX          ;incrementa el indice
        CMPB  #$05         ;cuando ingresaron 5 numeros quiere decir que se cargo la latitud
        BEQ   CLATF        ;por lo que envia a la rutina de latitud final

```

```

CMPB    #$0A      ;cuando ingresaron 10 numeros quiere decir que se cargo la longitud
BEQ     CLONF     ;por lo que envia a la rutina de longitud final
CMPB    #$0E      ;cuando ingresaron 14 numeros quiere decir que se cargo la altitud
BEQ     CALTF1    ;por lo que envia a la rutina de altitud final
JMP     SALESET   ;sale de la interrupcion

```

*RUTINA QUE PONE EL PROGRAMA EN RECARGA

```

RECAR   JSR     SAL_LIN      ;Salta a salto de linea
        JSR     RECA        ;escribe recarga en la pantalla
        LDX    #T1          ;resetea el indice x para guardar al principio de
                                ;los temporales de una cifra
        LDAB   #$00         ;resetea las bandera contador
        STAB   FLAG         ; inicializa las bandera contador
        JMP    SALESET      ;sale de la interrupcion

```

*RUTINA QUE TRANSFORMA LA LATITUD FINAL DE HEXADECIMAL A DECIMAL

```

CLATF   LDX    #T1          ;cambia el indice al coienzo de los temporales
        LDD    #$0000       ;borra el valor que del TEMP1
        STD    TEMP1        ;borra el valor que del TEMP1
        BSR    CAMBIAS      ;brinca a la subrutina que realiza el cambio de base para 5 cifras
        LDD    TEMP1        ;crga el resultado de la conversion
        STD    LATF         ;lo guarda en la localidad de latitud final
        LDAA   #' '        ;carga a A un " " en ASCII
        JSR    SERIAL       ;envia a la PC
        LDAA   #'L'        ;carga a A un "L" en ASCII
        JSR    SERIAL       ;envia a la PC
        LDAA   #'a'        ;carga a A un "a" en ASCII
        JSR    SERIAL       ;envia a la PC
        LDAA   #'t'        ;carga a A un "t" en ASCII
        JSR    SERIAL       ;envia a la PC
        JSR    SAL_LIN      ;Salta a salto de linea
        LDAB   #$05         ;pone 5 en la bandera
        JMP    SALESET      ;sale de la interrupcion

```

*PUENTES PARA QUE EL PROGRAMA ALCANCE LAS RUTINAS

```

SALED1  JMP     SALEDAT     ;puentea para que alcance la rutina SALEDAT
CALTF1  JMP     CALTF       ;puentea para que alcance la rutina CALTF
CAMBIAS JMP     CAMBIA      ;puentea para que alcance la rutina CAMBIA
CAMBIAS2 JMP    CAMBIA2     ;puentea para que alcance la rutina CAMBIA2
ERROR1  JMP     ERROR       ;puentea para que alcance la rutina ERROR

```

*RUTINA QUE TRANSFORMA LA LONGITUD FINAL DE HEXADECIMAL A DECIMAL

```

CLONF   LDX    #T2          ;cambia el indice al segundo valor de los temporales
        LDD    #$0000       ;para el cambio de un valor de 4 cifras
        STD    TEMP1        ;borra el valor que se del TEMP1
        BSR    CAMBIAS      ;brinca a la subrutina que realiza el cambio de base para 5 cifras
        LDD    TEMP1        ;crga el resultado de la conversion
        STD    LONF         ;lo guarda en la localidad de longitud final
        LDAA   #' '        ;carga a A un " " en ASCII
        JSR    SERIAL       ;envia a la PC
        LDAA   #'L'        ;carga a A un "L" en ASCII
        JSR    SERIAL       ;envia a la PC
        LDAA   #'o'        ;carga a A un "" en ASCII
        JSR    SERIAL       ;envia a la PC
        LDAA   #'n'        ;carga a A un "n" en ASCII
        JSR    SERIAL       ;envia a la PC
        JSR    SAL_LIN      ;Salta a salto de linea
        LDAB   #$A         ;pone 10 en la bandera
        JMP    SALESET      ;sale de la interrupcion

```

*RUTINA QUE TRANSFORMA LA ALTITUD FINAL DE HEXADECIMAL A DECIMAL

```

CALTF  LDX      #T1          ;cambia el indice al comienzo de los temporales
        LDD      #$0000      ;borra el valor que se del TEMP1
        STD      TEMP1
        BSR      CAMBIAS2    ;brinca a la subrutina que realiza el cambio de base para 4 cifras
        LDD      TEMP1      ;carga el resultado de la conversion
        STD      ALTF       ;lo guarda en la localidad de altitud final
        LDAA     #' '       ;carga a A un " " en ASCII
        JSR      SERIAL     ;envia a la PC
        JSR      SERIAL     ;envia a la PC
        LDAA     #'A'       ;carga a A un "A" en ASCII
        JSR      SERIAL     ;envia a la PC
        LDAA     #'l'       ;carga a A un "l" en ASCII
        JSR      SERIAL     ;envia a la PC
        LDAA     #'t'       ;carga a A un "t" en ASCII
        JSR      SERIAL     ;envia a la PC
        LDAB     #$00       ;resetea la bandera
        JSR      LO         ;envia a la rutina que escribe listo
        JSR      OO         ;envia a la rutina que escriba " o "
        JMP      RECAR      ;envia a a recarga

```

*RUTINA QUE ENVIA EL DATO AL PUERTO SERIE

```

SERIAL  STAA     SCDR,Y      ;envia al puerto serie
LOOP    BRCLR   SCSR,Y,$40,LOOP ;espera finalice la transmicion
        RTS

```

*RUTINA DE SALTO DE LINEA

```

SAL_LIN LDAA     #$0A      ;rutina que salta un renglon en la PC
        JSR      SERIAL
        LDAA     #$0D
        BSR      SERIAL
        RTS

```

* RUTINA QUE SALE DE LA INTERRUPCIOM 1*

```

SALESET STX      INDICE     ;guarda el indice
        STAB     FLAG      ;guarda la bandeja contador
        CLI      ;habilita interrupciones generales
        RTI      ;sale de interrupcion

```

**RECONFIGURA PSEUDOVECTOR DE INTERRUPCION *

```

SALEDAT LDAA     #$01      ;cambia la bandera de inicio para
        STAA     $FF,Y     ;que corra el programa de decision
        JSR      LO        ;escribe listo en la pantalla
        LDAA     #$7E     ;direccion del pseudovector de int.
        STAA     $C4      ;guarda en inicio del pseudovector
        LDX      #SER_INT  ;guarda direccionde subrutina
        STX      $C5      ;en el pseudovector
        LDX      #POSI     ;carga la posicion de donde va a guardar
        STX      INDICE    ;los datos de la cadena GPGGA
        CLI      ;habilita interrupciones generales
        RTI      ;sale de interrupcion

```


* RUTINA DE INTERRUPCION 2

```

SER_INT SEI                ;deshabilita int generales
        BCLR      SCSR,Y,$20 ;limpia la bandera de recepcion completa
        LDX      INDICE     ;carga el indice X
        LDAA     SCDR,Y     ;carga el dato recibido
        LDAB     #$FF      ;enciende el led de recibiendo datos
        STAB     $08,Y
        JMP      IDENT     ;envia a la rutina de guardado

```

* RUTINA QUE IDENTIFICA LA CADENA *

```

IDENT  LDAB     FLAG        ;verifica si ya se identifico la cadena de caracteres
        CMPB     #$04      ;si ya se identifico brinca
        BEQ     CHK2
        CMPA     #$47      ;compara el dato con G para identificar la trama
        BEQ     AS1        ;si es igual brinca
        CMPA     #$50      ;compara el dato con P para identificar la trama
        BEQ     AS2        ;si es igual brinca
S1     CMPA     #$47      ;compara el dato con G para identificar la trama
        BEQ     AS3        ;si es igual brinca
S2     CMPA     #$47      ;compara el dato con G para identificar la trama
        BEQ     AS4        ;si es igual brinca
        JMP     LIMPIA     ;pasa a la subrutina que limpia el indice y las banderas

```

* RUTINA QUE SALE DE LA INTERRUPCIOM *

```

SALE   STX      INDICE     ;guarda el indice
        CLI      ;habilita interrupciones generales
        RTI      ;sale de interrupcion

```

* RUTINA QUE GUARDA EL DATO RECIBIDO *

```

GUARDA STAA     $00,X      ;guarda el dato
        INX     ;incrementa el indice X
        JMP     SALE      ;manda a la rutina de identificacion

```

* LIMPIA LOS INDICES *

```

LIMPIA LDX      #POSI     ;limpia el indice X
        LDAB     #$00     ;limpia la bandera 1
        STAB     FLAG     ;guarda la bandera 1
        JMP     SALE      ;sale de la rutina de interrupcion

```

*RUTINAS DE ASIGNACION DE BANDERAS *

```

AS1    LDAB     FLAG      ;checa el valor de las banderas
        CMPB     #$00
        BNE     S1        ;si no tiene el mismo valor regresa
        LDAB     #$47
        STAB     POSI
        LDAB     #$01     ;cambia el valor de la bandera
        STAB     FLAG     ;guarda la bandera
        INX
        JMP     SALE      ;sale de interrupcion

AS2    LDAB     #$02     ;cambia el valor de la bandera
        STAB     FLAG     ;guarda la bandera
        JMP     GUARDA    ;sale de interrupcion

```

```

AS3   LDAB   FLAG           ;checa el valor de las banderas
      CMPB   #$02
      BNE    S2             ;si no tiene el mismo valor regresa
      LDAB   #$03           ;cambia el valor de la bandera
      STAB   FLAG           ;guarda la bandera
      JMP    GUARDA        ;sale de interrupcion

AS4   LDAB   FLAG           ;checa el valor de las banderas
      CMPB   #$03
      BNE    SALE           ;si no tiene el mismo valor regresa
      LDAB   #$04           ;cambia el valor de la bandera
      STAB   FLAG           ;guarda la bandera
      JMP    GUARDA        ;sale de interrupcion

CHK2  LDAA   SCDR,Y         ;carga el dato recibido
      CMPA   #$0A           ;compara el dato con el terminador
      BNE    GUARDA        ;si no es igual sale de la interrupcion
      LDX   #POSI           ;limpia el indice X
      LDAB   #$00           ;limpia la bandera 1
      STAB   FLAG           ;guarda la bandera 1
      JMP    UPDATE        ;salta a la rutina que separa los datos

*RUTINA QUE TRANSFORMA DE ASCII A HEXADECIMAL

TRANS LDAA   LA1            ;carga la primer cifra de la Latitud
      SUBA   #$30           ;transforma de ASCII a Hexa
      STAA   T1             ;la guarda el primer temporal
      LDAA   LA2            ;carga la segunda cifra de la Latitud
      SUBA   #$30           ;transforma de ASCII a Hexa
      STAA   T2             ;la guarda el segunda temporal
      LDAA   LA3            ;carga la tercera cifra de la Latitud
      SUBA   #$30           ;transforma de ASCII a Hexa
      STAA   T3             ;la guarda el tercera temporal
      LDAA   LA4            ;carga la cuarta cifra de la Latitud
      SUBA   #$30           ;transforma de ASCII a Hexa
      STAA   T4             ;la guarda el cuarta temporal
      LDAA   LA5            ;carga la quinta cifra de la Latitud
      SUBA   #$30           ;transforma de ASCII a Hexa
      STAA   T5             ;la guarda el quinta temporal
      LDD   #$0000         ;borra el valor que se del TEMP1
      STD   TEMP1
      BSR   CAMBIA         ;brinca a la subrutina que realiza el cambio de base para 5 cifras
      LDD   TEMP1          ;carga el resultado de la conversion
      STD   LATINS         ;lo guarda en la localidad de latitud instantanea

*CAMBIO DE LONGITUD
LDAA   LN1                ;carga la primer cifra de la Longitud
SUBA   #$30                ;transforma de ASCII a Hexa
STAA   T1                  ;la guarda el primer temporal
LDAA   LN2                ;carga la segunda cifra de la Longitud
SUBA   #$30                ;transforma de ASCII a Hexa
STAA   T2                  ;la guarda el segunda temporal
LDAA   LN3                ;carga la tercera cifra de la Longitud
SUBA   #$30                ;transforma de ASCII a Hexa
STAA   T3                  ;la guarda el tercera temporal
LDAA   LN4                ;carga la cuarta cifra de la Longitud
SUBA   #$30                ;transforma de ASCII a Hexa
STAA   T4                  ;la guarda el cuarta temporal
LDAA   LN5                ;carga la quinta cifra de la Longitud
SUBA   #$30                ;transforma de ASCII a Hexa
STAA   T5                  ;la guarda el quinta temporal
LDD   #$0000              ;borra el valor que se del TEMP1
STD   TEMP1
BSR   CAMBIA              ;brinca a la subrutina que realiza el cambio de base para 5 cifras
LDD   TEMP1               ;carga el resultado de la conversion
STD   LONINS              ;lo guarda en la localidad de longitud instantanea

*CAMBIO DE ALTURA
LDAA   AL1                ;carga la primer cifra de la altitud
SUBA   #$30                ;transforma de ASCII a Hexa
STAA   T2                  ;la guarda el segunda temporal puesto que la altitud solo tiene 4 cifras

```

```

LDAA    AL2    ;carga la segunda cifra de la altitud
SUBA    #$30   ;transforma de ASCII a Hexa
STAA    T3     ;la guarda el tercera temporal
LDAA    AL3    ;carga la tercera cifra de la altitud
SUBA    #$30   ;transforma de ASCII a Hexa
STAA    T4     ;la guarda el cuarta temporal
LDAA    AL4    ;carga la cuarta cifra de la altitud
SUBA    #$30   ;transforma de ASCII a Hexa
STAA    T5     ;la guarda el quinta temporal
LDD     #$0000 ;borra el valor que se del TEMP1
STD     TEMP1
BSR     CAMBIA2 ;brinca a la subrutina que realiza el cambio de base para 4 cifras
LDD     TEMP1  ;carga el resultado de la conversion
STD     ALTINS ;lo guarda en la localidad de altitud instantanea
JMP     DELTAS

```

*SUBRRUTINA QUE CAMBIA EL VALOR DECIMAL A HEXADECIMAL 5 CIFRAS

```

CAMBIA LDAA    T1     ;carga la primer cifra
LDAB    #$27   ;carga el factor de conversion de la parte alta de la primera cifra
MUL                      ;y los multiplica
STAB    TEMP1  ;guarda el valor resultante de la multiplicacion en la parte alta
LDAA    T1     ;carga la primer cifra
LDAB    #$10   ;carga el factor de conversion de la parte baja de la primera cifra
MUL                      ;y los multiplica
ADDD    TEMP1  ;suma el resultado a el valor obtenido anteriormente
STD     TEMP1  ;guarda el resultado obtenido

```

*SUBRRUTINA QUE CAMBIA EL VALOR DECIMAL A HEXADECIMAL 5 CIFRAS

```

CAMBIA2 LDAA    T2     ;carga la segunda cifra
LDAB    #$3    ;carga el factor de conversion de la parte alta de la segunda cifra
MUL                      ;y los multiplica
ADDB    TEMP1  ;suma el resultado a el valor obtenido anteriormente
STAB    TEMP1  ;guarda el resultado obtenido
LDAA    T2     ;carga la segunda cifra
LDAB    #$E8   ;carga el factor de conversion de la parte baja de la segunda cifra
MUL                      ;y los multiplica
ADDD    TEMP1  ;suma el resultado a el valor obtenido anteriormente
STD     TEMP1  ;guarda el resultado obtenido
LDAA    T3     ;carga la tercera cifra
LDAB    #$64   ;carga el valor del factor de conversion de la tercera cifra
MUL                      ;y los multiplica
ADDD    TEMP1  ;suma el resultado a el valor obtenido anteriormente
STD     TEMP1  ;guarda el resultado obtenido
LDAA    T4     ;carga la cuarta cifra
LDAB    #$A    ;carga el valor del factor de conversion de la cuarta cifra
MUL                      ;y los multiplica
ADDD    TEMP1  ;suma el resultado a el valor obtenido anteriormente
STD     TEMP1  ;guarda el resultado obtenido
LDAB    T5     ;carga la cuarta cifra
CLRA                      ;borra la parte alta del valor para solo sumar la cifra de las unidades
ADDD    TEMP1  ;suma la quinta cifra (unidades) al valor anterior
STD     TEMP1  ;guarda el resultado obtenido
RTS                      ;regresa de la subrutina

```

*RUTINA QUE CAMBIA LOS VALORES DE INSTANTANEOS A ANTERIORES

```

UPDATE LDD     LATINS  ;carga la latitud instantanea
STD     LATANT  ;y lo pasa a la latitud anterior
LDD     LONINS   ;carga la longitud instantanea
STD     LONANT  ;y lo pasa a la longitud anterior
LDD     ALTINS   ;carga la altitud instantanea
STD     ALTANT  ;y lo pasa a la altitud anterior
JMP     TRANS

```

*DELTA LATR LATITUD DE RUMBO latf-latins 16-bits

```

DELTAS LDD     LATF   ;carga la latitud final
STD     TEMP3  ;lo guarda en el temporal 3
LDD     LATINS   ;carga la latitud instantanea
STD     TEMP2  ;lo guarda en el temporal 2

```

```

JSR   RESTA   ;salta a la subrutina que realiza la resta
LDD   TEMP3   ;carga el resultado guardado temporalmente en el 3
STD   DELLR   ;lo guarda en el Delta de latitud de rumbo
LDAA  TEMP2   ;carga el dato del signo guardado temporalmente en 2
STAA  SDELLR  ;lo guarda en el signo del Delta de latitud de rumbo

*DELTA LnR LONGITUD DE RUMBO longf-longins 16-bits
LDD   LONF    ;carga la longitud final
STD   TEMP3   ;lo guarda en el temporal 3
LDD   LONINS  ;carga la longitud instantanea
STD   TEMP2   ;lo guarda en el temporal 2
BSR   RESTA   ;salta a la subrutina que realiza la resta
LDD   TEMP3   ;carga el resultado guardado temporalmente en el 3
STD   DELLnR  ;lo guarda en el Delta de longitud de rumbo
LDAA  TEMP2   ;carga el dato del signo guardado temporalmente en 2
STAA  SDELLnR ;lo guarda en el signo del Delta de longitud de rumbo

*DELTA LATD LATITUD DE DIRECCION latins-latant 16-bits
LDD   LATINS  ;carga la latitud instantanea
STD   TEMP3   ;lo guarda en el temporal 3
LDD   LATANT  ;carga la latitud anterior
STD   TEMP2   ;lo guarda en el temporal 2
BSR   RESTA   ;salta a la subrutina que realiza la resta
LDD   TEMP3   ;carga el resultado guardado temporalmente en el 3
STD   DELLD   ;lo guarda en el Delta de latitud de direccion
LDAA  TEMP2   ;carga el dato del signo guardado temporalmente en 2
STAA  SDELLD  ;lo guarda en el signo del Delta de latitud de direccion

*DELTA LnD LONGITUD DE DIRECCION longins-longant 16-bits
LDD   LONINS  ;carga la longitud instantanea
STD   TEMP3   ;lo guarda en el temporal 3
LDD   LONANT  ;carga la longitud anterior
STD   TEMP2   ;lo guarda en el temporal 2
BSR   RESTA   ;salta a la subrutina que realiza la resta
LDD   TEMP3   ;carga el resultado guardado temporalmente en el 3
STD   DELLnD  ;lo guarda en el Delta de longitud de direccion
LDAA  TEMP2   ;carga el dato del signo guardado temporalmente en 2
STAA  SDELLnD ;lo guarda en el signo del Delta de longitud de direccion

*DIFERENCIA DE ALTITUD ALTf-ALTINS
LDD   ALTf    ;carga la altitud final
STD   TEMP3   ;lo guarda en el temporal 3
LDD   ALTINS  ;carga la altitud instantanea
STD   TEMP2   ;lo guarda en el temporal 2
BSR   RESTA   ;salta a la subrutina que realiza la resta
LDD   TEMP3   ;carga el resultado guardado temporalmente en el 3
STD   DIFALT  ;lo guarda en la diferencia de altitudes
JMP   TAN     ;salta a la rutina que realiza las tangentes

*SUBRRUTINA QUE REALIZA LA RESTA DE NAGENTES

RESTA  LDD     TEMP3   ;carga el dato final
      CPD     TEMP2   ;lo compara con el inicial
      BHS     MAYOR   ;si el final es mayo brinca a MAYOR para realizar la resta
MENOR  LDD     TEMP2   ;si es menor los intercambia ya que el resultado va a ser negativo
      SUBD   TEMP3   ;ya que el resultado va a ser negativo
      STD     TEMP3   ;guarda el resultado en el temporal 3
      LDAA   #$01    ;carga el codigo del signo negativo
      STAA   TEMP2   ;y guarda el codigo del signo en el temporal 2
      RTS

MAYOR  SUBD   TEMP2   ;realiza la resta cuando el resultado va a ser positivo
      STD     TEMP3   ;guarda el resultado en el temporal 3
      CLRA   ;carga el codigo del signo positivo
      STAA   TEMP2   ;y guarda el codigo del signo en el temporal 2
      RTS

```

*REALIZA LAS TANGENTES

```

TAN      STX      INDSEPX1      ;guarda temporalmente el indice X
*PONE LOS DATOS PARA CALCULAR LA TANGENTE DE RUMBO
      LDD      DELLR      ;carga el delta de latitud de rumbo y lo guarda en
      STD      TEMP1      ;el temporal 1 para calcular la tangente
      LDD      DELLnR     ;carga el delta de longitud de rumbo y lo guarda en
      STD      TEMP2      ;el temporal 2 para calcular la tangente
      BSR      COSI       ;envia a la rutina que calcula la tangente
      LDD      T1         ;guarda el resultado de la tangente de rumbo
      STD      TANR

```

*PONE LOS DATOS PARA CALCULAR LA TANGENTE DE DIRECCION

```

      LDD      DELLD      ;carga el delta de latitud de direccion y lo guarda en
      STD      TEMP1      ;el temporal 1 para calcular la tangente
      LDD      DELLnD     ;carga el delta de longitud de direccion y lo guarda en
      STD      TEMP2      ;el temporal 2 para calcular la tangente
      BSR      COSI       ;envia a la rutina que calcula la tangente
      LDD      T1         ;guarda el resultado de la tangente de rumbo
      STD      TAND
      LDX      INDSEPX1   ;carga el valor del indice X

```

*DIFERENCIA DE TANGENTES

```

      LDD      TANR       ;carga la tangente de rumbo
      STD      TEMP3      ;lo guarda en el temporal 3
      LDD      TAND       ;carga la tangente de direccion
      STD      TEMP2      ;lo guarda en el temporal 2
      JSR      RESTA      ;salta a la subrutina que realiza la resta
      LDD      TEMP3      ;carga el resultado guardado temporalmente en el 3
      STD      DIFTAN     ;lo guarda en el Diferencia de tangentes
      JMP      SALE       ;Regresa a la tangente o cualquier otro lado del programa

```

* RUTINA QUE REALIZA EL COCIENTE

```

COSI     LDD      #$0000   ;limpia la posicion de memoria temporal
          STD      T1       ;
          LDD      TEMP1    ;carga el dato de latitud en el numerador
          LDX      TEMP2    ;carga el dato de la Longitud en el denominador
          IDIV     ;realiza el cociente y lo pone en X y el residuo en D
          XGDX    ;los intercambia y guarda la parte baja en el temporal de la
GRD_3    JMP      VERIF    ;verifica el numero de cifras del resultado de la divicion
          JSR      ALTOS    ;recorre el resultado a la izquierda a la primera posicion
          STAB     T1       ;primera cifra para convertirlo a Hexa
          XGDX    ;regresa el residuo a D
          BSR      MULTI    ;envia a multiplicarlo por 10 para calcular los decimales
          LDX      TEMP2    ;carga nuevamente la latitud en el denominador
          IDIV     ;vuelve a realizar el cociente entre el residuo x 10 y la latitud
          XGDX    ;intercambia el cociente con el resultado
GRD_2    JMP      VERIF2   ;verifica el numero de cifras del resultado de la divicion
          ADDB     T1       ;suma el resultado a la anterior en la parte alta
          STAB     T1       ;resultado en el temporal para conversion a Hexa
          XGDX    ;regresa el residuo a D
          BSR      MULTI    ;envia a multiplicarlo por 10 para calcular los decimales
LDX      TEMP2    ;carga nuevamente la latitud en el denominador
          IDIV     ;vuelve a realizar el cociente entre el residuo x 10 y la latitud
          XGDX    ;intercambia el cociente con resultado y guarda la parte baja
GRD_10   JMP      VERIF3   ;verifica el numero de cifras del resultado de la divicion
          JSR      ALTOS    ;recorre el resultado a la izquierda a la tercera posicion
          STAB     T2       ;resultado en el temporal para conversion a Hexa
          XGDX    ;regresa el residuo a D
          BSR      MULTI    ;envia a multiplicarlo por 10 para calcular los decimales
LDX      TEMP2    ;carga nuevamente la latitud en el denominador

```

```

        IDIV                ;vuelve a realizar el cociente entre el residuo x 10 y la latitud
        XGDY                ;intercambia el cociente con resultado y guarda la parte baja
ADDB   T2                  ;suma el resultado a la al anterior el la parte baja
        STAB   T2          ;resultado en el temporal para conversion a Hexa
SALECOS RTS                ;regresa de la subrutina

MULTI  CPD   #$100         ;checa si el resultado del residuo es de 2 digitos
        BLO   MULT2        ;brinca a la rutina de multiplicacion
        STD   TEMP1        ;guarda el numero a multiplicar en el temporal 1
        LDAB  #$00         ;borra el contenido de la parte baja del temporal 3
        STAB  TEMP3+1
*      LDD   TEMP1         ;carga el numero a multiplicar en D
        LDAB  #$0A         ;carga el 10 para multiplicarlo
        MUL   MUL          ;multiplica la parte alta por 10
        STAB  TEMP3        ;guarda el resultado en el temporal 3
        LDD   TEMP1        ;y vuelve a cargar el numero a multiplicar
        LDAA  #$0A         ;carga el 10 para multiplicarlo
        MUL   MUL          ;multiplica la parte baja por 10
        ADDD  TEMP3        ;y el resultado lo suma al resultado de la parte alta
SAL_MUL RTS                ;regresa de la subrutina

MULT2  JSR   ALTOS        ;agrega un cero al residuo
        RTS                ;sale de subrutina

*SALECOS RTS                ;sale de la subrutina

SALETAN LDX   INDSEPX1
        RTS

GRD_4  JSR   BAJOS        ;recorre el numero hacia la derecha ya que ya hay una cifra
GRD_5  ADDD  T1           ;suma el resultado a la parte alta
GRD_1  STD   T1           ;lo guarda en T1
*      JMP   SALECOS
        RTS                ;sale de la subrutina

GRD_6  JSR   ALTOS        ;corre el resultado a la izquierda para
        JMP   GRD_5        ;sumarlo a la cifra anterior

GRD_8  JSR   BAJOS        ;lo recorre una posicion a la derecha
GRD_7  STAA  T2           ;guarda la parte alta en la parte baja del resultado
*      JMP   SALECOS
        RTS                ;sale de la subrutina

GRD_9  STAB  T2           ;lo guarda en la parte baja del resultado
*      JMP   SALECOS
        RTS                ;sale de la subrutina

VERIF  CPD   #$100         ;verifica que la primer divicion cuando no hay cifras
        BHS   GRD_1        ;sea menor de 3 cifras si es asi lo guarda
        CPD   #$10         ;si es de 2 cifras guarda la parte alta
        BHS   GRD_2        ;si es de una cifra continua en el algoritmo de una cifra
        JMP   GRD_3

VERIF2 CPD   #$1000        ;verifica el resultado de la segunda divicion cuando ya hay una
cifra
        BHS   GRD_4        ;si es de 4 cifras va a recorrer el resultado va GRD_4
        CPD   #$100        ;si es de 3 cifras suna el resultado a la primera cifra
        BHS   GRD_5
        CPD   #$10         ;si es de 2 cifras va GRD_6
        BHS   GRD_6
        JMP   GRD_2        ;si es de una cirfra regresa al algoritmo original de 1 cifra

VERIF3 CPD   #$1000        ;verifica el resultado de la tercera divicion cuando ya hay dos
cifra
        BHS   GRD_7        ;si es de 4 cifras guarda la parte alta en la parte baja del
resultado
        CPD   #$100        ;si es de 3 cifras recorre el resultado una posicion a la derecha
        BHS   GRD_8        ;y lo guarda
        CPD   #$10

```

```

        BHS      GRD_9      ;si es de 2 cifras lo guarda en la parte baja del resultado
        JMP      GRD_10

ALTOS  LSLD                ;recorre el Acc D una cifra a la izquierda
        LSLD
        LSLD
        LSLD
        RTS

BAJOS  LSRD                ;recorre el Acc D una cifra a la derecha
        LSRD
        LSRD
        LSRD
        RTS

```

*RUTINA QUE ESCRIBE EN LA PANTALLA RECARGA

```

RECA   LDAA    #'R'      ;carga a A un "R" en ASCII
        JSR    SERIAL   ;envia a la PC
        LDAA    #'e'      ;carga a A un "e" en ASCII
        JSR    SERIAL   ;envia a la PC
        LDAA    #'c'      ;carga a A un "c" en ASCII
        JSR    SERIAL   ;envia a la PC
        LDAA    #'a'      ;carga a A un "a" en ASCII
        JSR    SERIAL   ;envia a la PC
        LDAA    #'r'      ;carga a A un "r" en ASCII
        JSR    SERIAL   ;envia a la PC
        LDAA    #'g'      ;carga a A un "g" en ASCII
        JSR    SERIAL   ;envia a la PC
        LDAA    #'a'      ;carga a A un "a" en ASCII
        JSR    SERIAL   ;envia a la PC
        JSR    SAL_LIN  ;Salta a salto de linea
        RTS

```

*RUTINA QUE ESCRIBE EN LA PANTALLA ERROR

```

ERROR  CLR     SCDR,Y
        JSR    SAL_LIN  ;Salta a salto de linea
        LDAA    #'E'      ;carga a A un "E" en ASCII
        JSR    SERIAL   ;envia a la PC
        LDAA    #'R'      ;carga a A un "R" en ASCII
        JSR    SERIAL   ;envia a la PC
        LDAA    #'R'      ;carga a A un "R" en ASCII
        JSR    SERIAL   ;envia a la PC
        LDAA    #'O'      ;carga a A un "O" en ASCII
        JSR    SERIAL   ;envia a la PC
        LDAA    #'R'      ;carga a A un "R" en ASCII
        JSR    SERIAL   ;envia a la PC
        LDAA    #' '      ;carga a A un " " en ASCII
        JSR    SERIAL   ;envia a la PC
        JMP    RECAR

```

*RUTINA QUE ESCRIBE EN LA PANTALLA LISTO

```

LO     JSR    SAL_LIN  ;Salta a salto de linea
        LDAA    #'L'      ;carga a A un "L" en ASCII
        JSR    SERIAL   ;envia a la PC
        LDAA    #'i'      ;carga a A un "i" en ASCII
        JSR    SERIAL   ;envia a la PC
        LDAA    #'s'      ;carga a A un "s" en ASCII
        JSR    SERIAL   ;envia a la PC
        LDAA    #'t'      ;carga a A un "t" en ASCII
        JSR    SERIAL   ;envia a la PC
        LDAA    #'o'      ;carga a A un "o" en ASCII
        JSR    SERIAL   ;envia a la PC
        RTS

```

*RUTINA QUE ESCRIBE EN LA PANTALLA " o "

```

OO      LDAA    #' '    ;carga a A un " " en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'o'    ;carga a A un "o" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #' '    ;carga a A un " " en ASCII
        JSR     SERIAL ;envia a la PC
        RTS

```

* RUTINA QUE ESCRIBE LA BIENVENIDA AL SISTEMA DE NAVEGACION

```

BIENV   JSR     SAL_LIN ;Salta a salto de linea
        LDAA    #'B'    ;carga a A un "B" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'I'    ;carga a A un "I" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'E'    ;carga a A un "E" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'N'    ;carga a A un "N" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'V'    ;carga a A un "V" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'E'    ;carga a A un "E" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'N'    ;carga a A un "N" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'I'    ;carga a A un "I" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'D'    ;carga a A un "D" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'O'    ;carga a A un "O" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #' '    ;carga a A un " " en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'A'    ;carga a A un "A" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'L'    ;carga a A un "L" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #' '    ;carga a A un " " en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'S'    ;carga a A un "S" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'I'    ;carga a A un "I" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'S'    ;carga a A un "S" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'T'    ;carga a A un "T" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'E'    ;carga a A un "E" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'M'    ;carga a A un "M" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'A'    ;carga a A un "A" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #' '    ;carga a A un " " en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'D'    ;carga a A un "D" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'E'    ;carga a A un "E" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #' '    ;carga a A un " " en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'N'    ;carga a A un "N" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'A'    ;carga a A un "A" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'V'    ;carga a A un "V" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'E'    ;carga a A un "E" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'G'    ;carga a A un "G" en ASCII
        JSR     SERIAL ;envia a la PC

```

```

LDAA    #'A'    ;carga a A un "A" en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #'C'    ;carga a A un "C" en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #'I'    ;carga a A un "I" en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #'O'    ;carga a A un "O" en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #'N'    ;carga a A un "N" en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #' '    ;carga a A un " " en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #'A'    ;carga a A un "A" en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #'U'    ;carga a A un "U" en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #'T'    ;carga a A un "T" en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #'O'    ;carga a A un "O" en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #'M'    ;carga a A un "M" en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #'A'    ;carga a A un "A" en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #'T'    ;carga a A un "T" en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #'A'    ;carga a A un "A" en ASCII
JSR     SERIAL ;envia a la PC
JSR     SAL_LIN ;Salta a salto de linea
RTS

```

* RUTINA QUE ESCRIBE LA CARGA DE DATOS

```

CODEST JSR     SAL_LIN ;Salta a salto de linea
        JSR     SAL_LIN ;Salta a salto de linea
LDAA    #'I'    ;carga a A un "I" en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #'N'    ;carga a A un "N" en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #'T'    ;carga a A un "T" en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #'R'    ;carga a A un "R" en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #'O'    ;carga a A un "O" en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #'D'    ;carga a A un "D" en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #'U'    ;carga a A un "U" en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #'S'    ;carga a A un "S" en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #'C'    ;carga a A un "C" en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #'A'    ;carga a A un "A" en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #' '    ;carga a A un " " en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #'L'    ;carga a A un "L" en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #'A'    ;carga a A un "A" en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #'S'    ;carga a A un "S" en ASCII
JSR     SERIAL ;envia a la PC
LDAA    #' '    ;carga a A un " " en ASCII
JSR     SERIAL ;envia a la PC
RTS

NUEV   LDAA    #'N'    ;carga a A un "N" en ASCII
        JSR     SERIAL ;envia a la PC
        LDAA    #'U'    ;carga a A un "U" en ASCII
        JSR     SERIAL ;envia a la PC

```

```

LDAA #'E' ;carga a A un "E" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'V' ;carga a A un "V" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'A' ;carga a A un "A" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'S' ;carga a A un "S" en ASCII
JSR SERIAL ;envia a la PC
LDAA #' ' ;carga a A un " " en ASCII
JSR SERIAL ;envia a la PC
RTS

COOR LDAA #'C' ;carga a A un "C" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'O' ;carga a A un "O" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'O' ;carga a A un "O" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'R' ;carga a A un "R" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'D' ;carga a A un "D" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'E' ;carga a A un "E" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'N' ;carga a A un "N" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'A' ;carga a A un "A" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'D' ;carga a A un "D" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'A' ;carga a A un "A" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'S' ;carga a A un "S" en ASCII
JSR SERIAL ;envia a la PC
LDAA #' ' ;carga a A un " " en ASCII
JSR SERIAL ;envia a la PC
LDAA #'D' ;carga a A un "D" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'E' ;carga a A un "E" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'L' ;carga a A un "L" en ASCII
JSR SERIAL ;envia a la PC
LDAA #' ' ;carga a A un " " en ASCII
JSR PTDEST
JSR SAL_LIN ;Salta a salto de linea
JSR SAL_LIN ;Salta a salto de linea
RTS

PTDEST JSR SERIAL ;envia a la PC
LDAA #'P' ;carga a A un "P" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'U' ;carga a A un "U" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'N' ;carga a A un "N" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'T' ;carga a A un "T" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'O' ;carga a A un "O" en ASCII
JSR SERIAL ;envia a la PC
LDAA #' ' ;carga a A un " " en ASCII
JSR SERIAL ;envia a la PC
LDAA #'D' ;carga a A un "D" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'E' ;carga a A un "E" en ASCII
JSR SERIAL ;envia a la PC
LDAA #' ' ;carga a A un " " en ASCII
JSR SERIAL ;envia a la PC
LDAA #'D' ;carga a A un "D" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'E' ;carga a A un "E" en ASCII
JSR SERIAL ;envia a la PC

```

```

LDAA #'S' ;carga a A un "S" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'T' ;carga a A un "T" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'I' ;carga a A un "I" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'N' ;carga a A un "N" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'O' ;carga a A un "O" en ASCII
JSR SERIAL ;envia a la PC
RTS

FINAL JSR SAL_LIN ;Salta a salto de linea
JSR SAL_LIN ;Salta a salto de linea
LDAA #'S' ;carga a A un "S" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'E' ;carga a A un "E" en ASCII
JSR SERIAL ;envia a la PC
LDAA #' ' ;carga a A un " " en ASCII
JSR SERIAL ;envia a la PC
LDAA #'H' ;carga a A un "H" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'A' ;carga a A un "A" en ASCII
JSR SERIAL ;envia a la PC
LDAA #' ' ;carga a A un " " en ASCII
JSR SERIAL ;envia a la PC
LDAA #'A' ;carga a A un "A" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'L' ;carga a A un "L" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'C' ;carga a A un "C" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'A' ;carga a A un "A" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'N' ;carga a A un "N" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'Z' ;carga a A un "Z" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'A' ;carga a A un "A" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'D' ;carga a A un "D" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'O' ;carga a A un "O" en ASCII
JSR SERIAL ;envia a la PC
LDAA #' ' ;carga a A un " " en ASCII
JSR SERIAL ;envia a la PC
LDAA #'E' ;carga a A un "E" en ASCII
JSR SERIAL ;envia a la PC
LDAA #'L' ;carga a A un "L" en ASCII
JSR SERIAL ;envia a la PC
LDAA #' ' ;carga a A un " " en ASCII
JSR SERIAL ;envia a la PC
JSR PTDEST
JSR SAL_LIN ;Salta a salto de linea
JSR SAL_LIN ;Salta a salto de linea
RTS

```