



**UNIVERSIDAD NACIONAL AUTONOMA DE MÉXICO**

---

FACULTAD DE INGENIERIA

DISEÑO DE UN SISTEMA DE CONTROL DE NAVEGACION  
AUTONOMA PARA UN VEHICULO MARITIMO BASADO EN  
EL SISTEMA DE POSICIONAMIENTO GLOBAL GPS  
UTILIZANDO EL MICROPROCESADOR M68HC11.

**T E S I S**

QUE PARA OBTENER EL TITULO DE:  
**INGENIERO EN TELECOMUNICACIONES**  
P R E S E N T A N :  
**ALFONSO ROJAS DOMINGUEZ**  
**CARLOS ROMO FUENTES**

DIRECTOR M. EN I. ARTURO GONZALEZ HERMOSILLO MELGAREJO

MEXICO, D.F.

JUNIO DE 2002

**TESIS CON  
FALLA DE ORIGEN**



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## **AGRADECIMIENTOS**

---

Al Ing. Arturo González Hermosillo por darnos la oportunidad de trabajar a su lado.

A los amigos que nos dejó la carrera de Ingeniería de Telecomunicaciones.

A la Universidad Nacional Autónoma de México, por sus maestros y su conocimiento.

### *Alfonso*

A mi familia y amigos, que constantemente inspiran mi superación y me dan confianza en todos los aspectos de mi vida.

A mi padre, de quien siempre he tenido su apoyo para mis estudios, su amor y su experiencia.

A Óscar, Carlos y Frank, que aportaron buenas ideas, esfuerzo y dedicación al desarrollo de este trabajo y con quienes disfruto enormemente los momentos de ocio.

### *Carlos Romo Fuentes*

A mis padres, mamá Inés y papá Carlos, por su apoyo durante mis estudios, su confianza y consejos, y sobre todo por su amor incondicional que me han dado durante toda mi vida.

A mis hermanas, Fabiola y Tatiana, cuñado y sobrino, Anibal y Anibalín, que me han alentado a seguir estudiando y me han dado razones para seguir preparándome.

A toda mi familia, primos y tíos, que siempre han estado en las buenas y en las malas y sobreponiéndonos de todas las caídas.

A mi amigo Poncho por aguantarme todo este tiempo en el que he aprendido mucho de él.

A mis amigos del Logos que sé que siempre estarán conmigo.

# ÍNDICE

|  |           |
|--|-----------|
| <b>1 INTRODUCCIÓN</b>  | <b>1</b>  |
| 1.1 LOS SISTEMAS AUTÓNOMOS DE NAVEGACIÓN                     | 1         |
| 1.2 OBJETIVOS DEL PROYECTO                                   | 1         |
| 1.3 DESCRIPCIÓN GENERAL DEL PROYECTO                         | 2         |
| 1.4 JUSTIFICACIÓN Y ALCANCES DEL PROYECTO                    | 2         |
| <b>2 SISTEMA DE POSICIONAMIENTO GLOBAL GPS</b>               | <b>5</b>  |
| 2.1 FORMACIÓN DEL SISTEMA                                    | 5         |
| 2.1.1 Segmento espacial                                      | 6         |
| 2.1.2 Segmento de control                                    | 6         |
| 2.1.3 Segmento del usuario                                   | 7         |
| 2.2 USOS DEL SISTEMA GPS                                     | 7         |
| 2.3 FUNCIONAMIENTO DEL GPS                                   | 8         |
| 2.3.1 El método de triangulación                             | 8         |
| 2.3.2 Triangulación aplicada en el GPS                       | 10        |
| 2.4 CARACTERÍSTICAS DE LA SEÑAL GPS                          | 13        |
| 2.5 EL RECEPTOR GPS  | 15        |
| 2.5.1 Elementos esenciales de un receptor GPS                | 15        |
| 2.5.2 Especificaciones técnicas del receptor GPS utilizado   | 17        |
| 2.6 EL PROTOCOLO DE COMUNICACIÓN NMEA 0183                   | 20        |
| 2.6.1 El estándar NMEA 0183                                  | 20        |
| 2.6.2 Interfaz eléctrica                                     | 20        |
| 2.6.3 Formato general de los mensajes                        | 21        |
| 2.6.4 Formato del mensaje GPS                                | 21        |
| <b>3 EL MICROPROCESADOR M68HC11</b>                          | <b>23</b> |
| 3.1 EL MICROPROCESADOR                                       | 23        |
| 3.2 ESPECIFICACIONES TÉCNICAS DEL MICROPROCESADOR M68HC11    | 24        |
| 3.3 LOS PUERTOS Y REGISTROS DEL MICROPROCESADOR              | 26        |
| 3.3.1 Puertos del microprocesador                            | 26        |
| 3.3.2 Registros del microprocesador                          | 26        |
| 3.4 INTERFAZ DE COMUNICACIÓN SERIAL                          | 28        |
| 3.5 MEMORIA DEL MICROPROCESADOR                              | 28        |
| <b>4 SUBSISTEMAS</b>   | <b>30</b> |
| 4.1 SENSORES   | 30        |
| 4.1.1 Teoría de sensores                                     | 30        |
| 4.1.2 Tipos de sensores                                      | 32        |
| 4.1.3 Funcionamiento de los sensores ultrasónicos            | 34        |
| 4.1.4 Características de los sensores ultrasónicos empleados | 37        |
| 4.2 SISTEMAS DE PROPULSIÓN Y DIRECCIÓN                       | 40        |
| 4.2.1 Control del motor de propulsión del vehículo           | 41        |
| 4.2.2 Control del motor de dirección del vehículo            | 45        |



|  |           |
|--|-----------|
| <b>5 ALGORITMOS DE NAVEGACIÓN AUTÓNOMA</b>                       | <b>47</b> |
| 5.1 POSICIONAMIENTO  | 48        |
| 5.2 PLANIFICACIÓN DE TRAYECTORIAS                                | 49        |
| 5.3 ALGORITMO DE NAVEGACIÓN PARA UN VEHÍCULO MARÍTIMO            | 51        |
| 5.3.1 Estructura del algoritmo de navegación marítima            | 52        |
| <b>6 INTEGRACIÓN DEL SISTEMA</b>                                 | <b>60</b> |
| 6.1 PROTOCOLO DE ENCENDIDO DEL SISTEMA                           | 60        |
| 6.2 INICIALIZACIÓN DEL PROGRAMA                                  | 61        |
| 6.3 AUTONOMÍA DEL SISTEMA  | 62        |
| 6.3.1 Toma de lecturas del receptor GPS                          | 62        |
| 6.3.2 Rastreo de posición y corrección de trayectoria            | 63        |
| 6.3.3 Interrupción por sensores                                  | 70        |
| 6.4 IMPLEMENTACIÓN DE PROTOTIPOS                                 | 71        |
| <b>7 PRUEBAS Y RESULTADOS</b>                                    | <b>75</b> |
| 7.1 PRUEBAS DE MONITOREO DEL SISTEMA                             | 75        |
| 7.1.1 Pruebas de rastreo de trayectorias                         | 75        |
| 7.1.2 Pruebas de precisión estática                              | 78        |
| 7.2 PRUEBAS DE DESEMPEÑO BAJO CONDICIONES CONTROLADAS            | 82        |
| 7.2.1 Prueba de identificación de casos                          | 82        |
| 7.2.2 Prueba de recorrido de trayectoria fija                    | 83        |
| 7.3 PRUEBAS DE DESEMPEÑO AUTÓNOMO                                | 84        |
| 7.3.1 Prueba de recorrido de trayectoria indicada por el sistema | 84        |
| 7.3.2 Prueba de verificación del límite de acercamiento          | 88        |
| <b>8 CONCLUSIONES</b>  | <b>91</b> |
| <b>APÉNDICE</b>  | <b>93</b> |

## ***Introducción***

### ***1.1 LOS SISTEMAS AUTÓNOMOS DE NAVEGACIÓN***

Actualmente existe una tendencia a crear sistemas autónomos para optimizar procesos en los cuales la intervención humana, gracias al desarrollo de la tecnología, ya no es indispensable; tal es el caso de las plantas de ensamblaje o las fábricas procesadoras y empacadoras de alimentos. También existen sistemas robóticos más complejos, diseñados para tareas tan diversas como son la inspección de ductos en busca de fugas o la excavación de minas.

Con estos sistemas se evita que el ser humano realice tareas que resultan peligrosas, complicadas o tediosas. La automatización de procesos es un área que demanda gran especialización y en la que convergen distintas disciplinas: las ingenierías; el control de calidad; la economía; la biología; etc. Los sistemas automáticos en general requieren de mucho tiempo de diseño y desarrollo, además de profundos estudios sobre las aplicaciones específicas a las que van dirigidos. Sin embargo, a largo plazo su relación de costo - beneficio siempre es positiva, por lo que en general se decide invertir en este tipo de sistemas.

Por otra parte, las comunicaciones via satélite han ganado importancia en una gran parte de las áreas de investigación respecto de los sistemas de comunicación alámbricos, sobre todo en aquellas áreas en las que la transmisión de datos a grandes distancias es primordial para su desarrollo. Muchos sistemas satelitales para comunicación (telefonía inalámbrica, internet móvil, televisión digital) y localización (LORAN, GPS, GLONASS) han sido desarrollados a lo largo de las últimas décadas.

El Sistema de Posicionamiento Global (GPS) aporta, entre otros, datos certeros de la localización de móviles sobre la superficie terrestre. Este sistema se encuentra activo las 24 horas del día. Gracias a esto, puede pensarse en el diseño de un vehículo marítimo que navegue de manera autónoma de un punto a otro del planeta guiándose por el sistema GPS. El proyecto que se presenta en este documento se centra en la investigación y el desarrollo de tecnología para uno de los componentes principales de esta idea (el sistema de control de navegación), integrando en el proceso varios aspectos de las ingenierías eléctrica-electrónica, de computación y de telecomunicaciones.

## **1.2 OBJETIVOS DEL PROYECTO**

Dos fueron los objetivos principales a cumplir con este proyecto:

- 1) Diseñar el modelo de un sistema de control de navegación autónoma para un vehículo marítimo con base en el sistema satelital de posicionamiento global (GPS). Este sistema de control debe contener todos los elementos necesarios para proveer a un vehículo marítimo de pequeñas dimensiones (2 o 2.5 (m) de envergadura) con la capacidad de realizar las tareas fundamentales que definen la navegación: determinación de la posición del móvil; percepción del entorno; planificación de la trayectoria del móvil; y movimiento del móvil a través de su entorno.
- 2) Realizar un documento que sirva como referencia bibliográfica para proyectos relacionados con el control de navegación autónoma de vehículos marítimos.

Se presenta este escrito con la intención de explicar el desarrollo del proyecto de tesis para licenciatura en *Ingeniería en Telecomunicaciones* de los autores y cumplir, a la vez, con el segundo de los objetivos enumerados.

Este documento consta de ocho capítulos y un apéndice. En el primer capítulo se expone la estructura del escrito y se ofrece una breve descripción de la manera en que se desarrolló el proyecto. En los capítulos dos al cuatro se describen detalladamente los componentes que integran el sistema de navegación desarrollado, incluyendo las bases teóricas de los sistemas y las especificaciones técnicas de los dispositivos utilizados.

El capítulo cinco presenta el algoritmo de navegación diseñado. Éste constituye la parte lógica del sistema, por lo que se explica en la forma más sencilla posible, sin profundizar en su programación. Sin embargo, en el apéndice se encuentran los listados del programa completo del algoritmo de navegación y del resto del sistema, explicados con detalle.

En el capítulo seis se aborda la integración del sistema, los procedimientos para la puesta en marcha y el funcionamiento del mismo tratado ya como una unidad. En el capítulo siete se reportan las pruebas de desempeño realizadas sobre el sistema y los resultados que se obtienen. Finalmente, el capítulo ocho presenta las conclusiones finales sobre el sistema en general.

## **1.3 DESCRIPCIÓN GENERAL DEL PROYECTO**

El Sistema de Control de Navegación se desarrolló en cuatro grandes etapas. Durante las tres primeras etapas se trabajó bajo condiciones de laboratorio con la finalidad de preparar el sistema para la etapa final que consistió en la realización de pruebas de campo.

**Primera etapa:** Esta etapa se enfocó al diseño y la implementación de los circuitos de control para los motores de propulsión y dirección del móvil. Se implementaron prototipos de estos sistemas y se programaron los controladores por software correspondientes. Esto se realizó cuidando que el sistema tuviera la capacidad para trabajar con otros equipos (motores con mayor o menor potencia) sin la necesidad de modificar el programa de control.

Se realizaron pruebas de control de los motores sobre los prototipos, y una vez que se dispuso de una configuración confiable de los elementos de hardware se prosiguió a la segunda etapa.

**Segunda Etapa:** La segunda etapa consistió en resolver el problema de la adquisición de datos de la señal GPS. Se diseñó una interfaz (de hardware y de software) para comunicar un receptor GPS con el microcontrolador M68HC11. Mediante esta interfaz fue posible el monitoreo y manipulación de la información de posición y trayectoria del móvil (del receptor GPS).

Las herramientas utilizadas en esta etapa fueron: la programación del microprocesador; el manejo del receptor GPS; y simulaciones por software realizadas de acuerdo con las especificaciones y recomendaciones de la norma NMEA0183 dirigida a la comunicación entre instrumentación marina, la cual ayudó a relacionar estos dos dispositivos.

**Tercera Etapa:** El objetivo de esta etapa fue la integración de los elementos desarrollados en las dos etapas anteriores. Esto se logró mediante la creación del programa de control del sistema, y la creación y programación del algoritmo para la navegación del móvil. Esta fue la etapa más compleja y larga del proyecto, pues requirió de mucho tiempo para la programación del algoritmo además del dedicado a la verificación del funcionamiento simultáneo de los elementos del sistema y a la depuración de los programas.

**Cuarta Etapa:** En la última etapa del proyecto se realizaron pruebas diseñadas para evaluar el desempeño general del sistema y justificar el buen funcionamiento del mismo teniendo como base los resultados obtenidos. Las pruebas realizadas pueden dividirse en tres tipos:

- 1) **Monitoreo del sistema:** Trazo de mapas de trayectoria a partir de coordenadas recopiladas por el sistema en movimiento. Su fin fue verificar el desempeño del receptor GPS y evaluar su precisión en el posicionamiento experimentalmente.
- 2) **Pruebas de desempeño del sistema sobre trayectorias previamente fijadas.** Su fin fue comprobar que los cálculos y procesos que el sistema realiza fueran correctos.
- 3) **Pruebas de desempeño autónomo.** En estas pruebas se expuso el sistema a condiciones que se asemejan a sus condiciones de funcionamiento real (trayectorias sin limitaciones de movimiento y con respuestas en tiempo real a las instrucciones del sistema).

## ***1.4 JUSTIFICACIÓN Y ALCANCES DEL PROYECTO***

Este proyecto adquiere su justificación derivada de la necesidad que presentan algunos sectores de la industria y las instituciones dedicadas a la investigación científica de adquirir datos (mediciones y muestras) sobre cuerpos acuáticos que muchas veces resultan de difícil acceso o que representan alto riesgo y costos elevados. Por ejemplo, cuando ocurren derrames de petróleo u otras sustancias que resultan tóxicas para el ser humano, o donde existan características climáticas agresivas (bajas temperaturas, condiciones de iluminación nula, etc.). Además, el sistema de navegación desarrollado puede utilizarse en muchas otras aplicaciones. Una perspectiva a futuro se presenta en el último capítulo de este escrito como parte de las conclusiones.

El sistema que se propone representa la iniciativa para el desarrollo de tecnología en el área de control y automatización enfocada a colaborar en estas tareas riesgosas o tediosas. Adicionalmente, este sistema se logra implementar con un costo reducido, en comparación con sistemas semejantes que utilizan un equipo más complejo, como cámaras de video en lugar de sensores ultrasónicos, microprocesadores con mayor capacidad de procesamiento, maquinaria de precisión, etc.

Este proyecto se concentra en el control de vehículos marítimos, no pretende cubrir los aspectos relativos a vehículos con movimiento en tres dimensiones (vehículos aéreos o submarinos), ni de vehículos terrestres que cuentan o requieren de mayor precisión en el recorrido de sus trayectorias. El sistema desarrollado fue diseñado para desempeñarse en ambientes que no presenten grandes perturbaciones (lagunas, mar en calma, canales, etc.) ni demasiada concentración de obstáculos durante la trayectoria del vehículo. El sistema tiene algunas limitaciones, como son: la complejidad de los algoritmos de control al momento de traducirlos de su expresión matemática a lenguaje máquina; la disponibilidad de los satélites GPS que no siempre se encuentran a la vista del receptor; la precisión de los sensores utilizados, para obtener las lecturas adecuadas sin ninguna confusión; las capacidades del microprocesador empleado (como la longitud de sus registros y sus operaciones definidas); y el tipo de vehículo que se controla, un vehículo marítimo de pequeñas dimensiones.

Aunque la comunicación del sistema con el exterior es unidireccional (del sistema satelital al receptor GPS), existe la posibilidad de implementar un canal de comunicación entre el vehículo y una estación terrestre que realice el monitoreo de su posición, reporte los datos recabados por el mismo y actualice las nuevas trayectorias a seguir. Como se verá a lo largo de este escrito, el diseño del sistema es muy flexible, sus componentes sencillos y de fácil implementación; además, los recursos del microprocesador no fueron rebasados. Todo esto permite pensar en una gran variedad de aplicaciones distintas que pueden lograrse sobre el diseño base.

**2*****Sistema de Posicionamiento Global GPS*****2.1 FORMACIÓN DEL SISTEMA**

GPS son las siglas de *Global Positioning System*, una constelación de 24 satélites que orbitan la tierra dos veces por día a una altitud de aproximadamente 12,000 millas o 20,000 Km. Las 24 horas del día estos satélites transmiten señales de radio de alta frecuencia que contienen información de tiempo y posición, así como datos de las efemérides y el almanaque de los satélites. Esto permite que cualquiera que posea un receptor GPS pueda determinar su localización en cualquier punto de la tierra que se encuentre al aire libre. Las señales del GPS están disponibles para un número ilimitado de usuarios simultáneamente.

El almanaque es un conjunto de parámetros usados por un receptor GPS para predecir la localización aproximada de un satélite GPS y la deriva esperada del reloj del satélite respecto a su reloj interno. Cada satélite GPS contiene y transmite los datos del almanaque de todos los satélites.

Las efemérides son otro conjunto de parámetros usados por un receptor GPS para predecir la localización de un satélite GPS y el comportamiento de su reloj. Cada satélite contiene y transmite datos de efemérides concernientes a su propia órbita y reloj. Los datos de las efemérides son más precisos que los del almanaque, pero son aplicables durante un tiempo más corto (de cuatro a seis horas). Los datos de efemérides se transmiten por cada satélite cada 30 segundos.

El departamento de defensa de Estados Unidos comenzó el desarrollo del sistema GPS en los 70s, con el fin de proveer información de posición y navegación a escala global y de forma ininterrumpida a las fuerzas militares de ese país. Sin embargo, al paso del tiempo se han encontrado un gran número de aplicaciones para el GPS en el ámbito civil. La información de posicionamiento y navegación que ofrece el GPS es vital para muchas actividades personales y profesionales como la aviación, exploración, rastreo de vehículos, etc. La base de la tecnología GPS es el procesamiento de información de tiempo y posición de una forma precisa. Usando relojes atómicos (con una precisión de alrededor de un segundo por cada 70,000 años) y datos de localización, cada satélite transmite continuamente el tiempo u hora actual y su posición.

Para describir el Sistema de Posicionamiento Global, se sigue la división acostumbrada para los sistemas satelitales de navegación en tres segmentos: **segmento espacial** que se refiere a la constelación de satélites; **segmento de control** que monitorea y controla todo el sistema; y **segmento del usuario** que consiste en los distintos tipos de receptores. A continuación se da una breve descripción de cada uno de estos segmentos.

### 2.1.1 Segmento Espacial

La constelación del sistema y su número total de satélites han sufrido variaciones con el tiempo. Los primeros satélites GPS en órbita tenían una inclinación de  $63^\circ$  con respecto al Ecuador y los planes de ese entonces eran colocar 24 satélites en 3 planos orbitales. Debido a cuestiones presupuestarias la constelación se pensó reducir a 18 satélites. Con esta idea, sin embargo, no se proveía la cobertura deseada. La constelación final de satélites GPS se estableció en 21 satélites principales más tres satélites inactivos de repuesto orbitando la tierra en órbitas casi circulares a una elevación de aproximadamente 20,000 Km sobre la tierra y con un periodo de 12 horas sidéreas. Estos satélites tienen una inclinación de  $55^\circ$  con respecto al Ecuador y están colocados en seis planos equidistantes con 4 satélites en cada órbita. La separación de los planos de las órbitas es de  $60^\circ$  en ascensión recta. La figura 2.1 ilustra esta configuración.

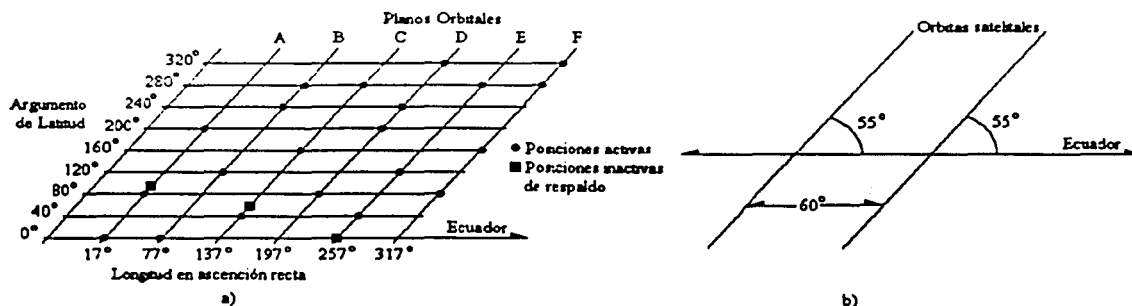


Figura 2.1 a) Representación Simplificada de la Constelación GPS.  
b) Separación Orbital e Inclinación de las Órbitas Respecto al Ecuador.

El Sistema de Posicionamiento Global actualmente emplea 24 satélites para garantizar que al menos 5 de ellos estén en el horizonte de cualquier punto sobre la tierra en cualquier momento. En términos generales siempre existen ocho o más satélites visibles para un receptor GPS. Este hecho se llama Capacidad de Operación Inicial (COI) [ 1].

### 2.1.2 Segmento De Control

Este segmento consiste de una red de estaciones para controlar y actualizar el sistema de satélites. Esto se logra mediante el constante monitoreo de los satélites desde una serie de estaciones convenientemente ubicadas alrededor de la tierra. Existe una estación de control maestra en el Centro de Operaciones Consolidadas del Espacio, en Colorado Springs. En esta estación se reúne la información de las estaciones de monitoreo y con estos datos se calculan las órbitas de los satélites y correcciones a los relojes, haciendo uso de estimadores Kalman [2]. 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 Índico 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.

Las estaciones son usadas para la determinación de las efemérides transmitidas y modelados de reloj. Las correcciones a las órbitas y relojes son retroalimentadas a los satélites mediante las estaciones de control terrestres. Estas estaciones se encuentran en Ascención, Diego García y Kwajalein.

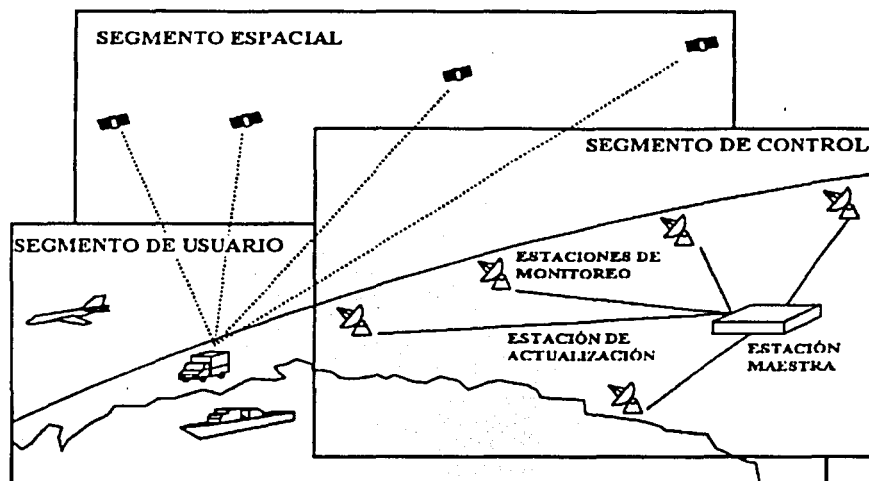


Figura 2.2 Sistema NAVSTAR-GPS: Segmento Espacial, Segmento de Control, Segmento de Usuario

### 2.1.3 Segmento Del Usuario

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

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

## 2.2 USOS DEL SISTEMA GPS

La idea original del GPS, que aun hoy día se mantiene, era usarlo para navegación. Esto es, conocer la posición del receptor en cualquier momento del día dentro de un sistema de referencia creado para tal fin. Esto es conocido como posicionamiento absoluto. Existen 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. Tiempo después, en mayo del año 2000, el gobierno de Estados Unidos deshabilitó la modalidad de Disponibilidad Selectiva de las señales GPS, lo que resultó en 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<sup>4</sup>. Los usuarios del sistema GPS pueden incrementar aún más la precisión de los datos mediante el uso de un método llamado GPS Diferencial (*DGPS* por sus siglas en inglés), o de otros sistemas satelitales como el *Wide Area Augmentation System (WAAS)*, el sistema europeo EGNOS o el *Multifunctional 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 [3].

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 (m). 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. Así, 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 han cartografiado cada esquina de la Tierra, así que pueden encontrarse mapas con el nivel de detalle que se prefiera.

## 2.3 FUNCIONAMIENTO DEL GPS

### 2.3.1 El Método De Triangulación

El sistema NAVSTAR-GPS se basa en la medida de la distancia entre el receptor y los satélites de la constelación GPS. Para calcular su posición es necesario que el receptor conozca su posición relativa respecto de los satélites, es decir, la posición de cada satélite y la distancia a la que el receptor se encuentra de ellos. A partir de esa información, el receptor aplica el concepto de triangulación y así puede encontrar su posición [4].

La triangulación o 'trilateración' es el método que permite encontrar la posición de un punto en el espacio a partir de distancias medidas desde ese punto a un número determinado de posiciones conocidas. Por lo general este método se aplica en un espacio de dos dimensiones (el plano) donde se necesitan tres posiciones conocidas para determinar la posición desconocida (por ello el nombre de 'triangulación'), sin embargo, el mismo principio puede usarse en espacios de una o tres dimensiones. Por ejemplo, considérese el caso de una dimensión, en la que se busca determinar la posición de un punto P sobre una recta. Se conocen las distancias  $d_1$  y  $d_2$ , medidas sobre la recta desde P hasta dos puntos  $q_1$  y  $q_2$  respectivamente. La figura 2.3 ilustra esta situación.

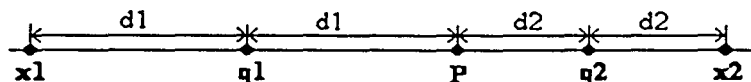


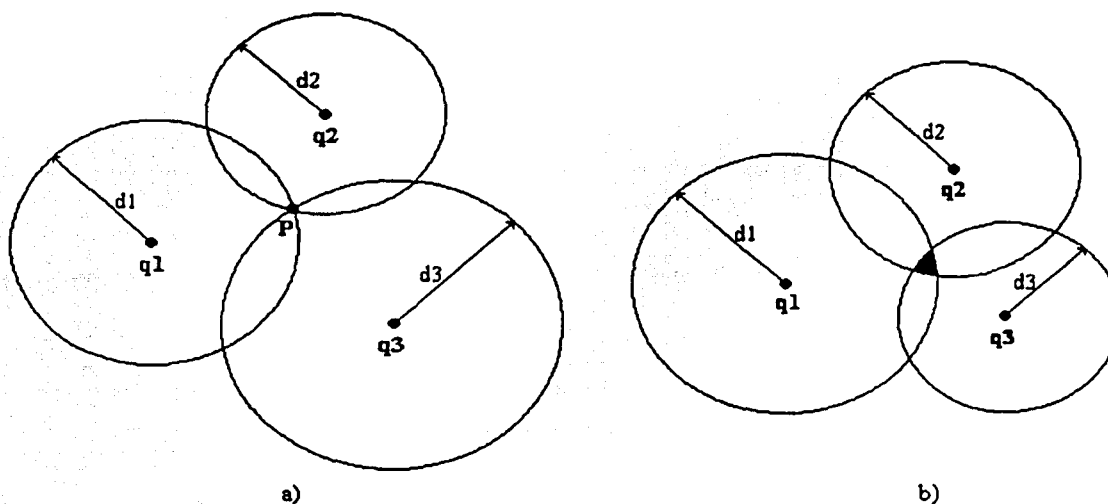
Figura 2.3 Caso de 'Triangulación' en un Espacio de una Dimensión

<sup>4</sup> La precisión del sistema GPS no ha sido reportada con exactitud, pues depende de la tecnología utilizada en cada receptor. En la mayoría de la bibliografía consultada, este dato oscila entre 8 y 10 (m).

Desde el punto  $q_1$ , la distancia  $d_1$  define dos puntos sobre la recta,  $x_1$  y  $P$ . A su vez, desde  $q_2$  la distancia  $d_2$  define otro par de puntos,  $x_2$  y  $P$ . Si las distancias  $d_1$  y  $d_2$  han sido medidas correctamente (lo que es el caso ideal), uno de los puntos medidos desde  $q_2$  coincidirá con uno de los puntos medidos desde  $q_1$ . El punto de coincidencia es, lógicamente, la posición buscada  $P$ .

Para el caso de dos dimensiones se requiere conocer tres posiciones y tres distancias medidas desde éstas a la posición buscada. Cada distancia define una circunferencia de posibles posiciones del punto buscado. El punto en donde se intersequen las tres circunferencias es la solución a la posición. La figura 2.4 a) ilustra este caso.

Si las distancias medidas desde el punto buscado hasta las posiciones de referencia carecen de precisión, las circunferencias trazadas no se intersecarán 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 es el buscado (figura 2.4 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. Si se conoce un número mayor de posiciones de referencia y distancias medidas a esos puntos, el método de aproximaciones convergerá más rápido. La geometría que presenten los puntos de referencia (la posición relativa entre ellos) también influye en la determinación de la solución. Con una buena geometría, aunque haya error en la medición de las distancias, se obtendrá un área de solución pequeña de la que será fácil obtener el punto buscado.



**Figura 2.4** 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.

En el caso de tres dimensiones, como puede inferirse, se requiere conocer cuatro posiciones y las correspondientes distancias medidas al 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 intersecan en una circunferencia. Con tres esferas se obtienen dos posibles puntos de solución. La cuarta esfera determina unívocamente la posición buscada.

De nuevo, si la medición de las distancias desde cada punto de referencia carece de precisión, la solución no será un solo punto, sino un conjunto de ellos. La posición del punto buscado deberá aproximarse hasta alcanzar la precisión deseada. En la siguiente sección se analiza más detalladamente el caso de triangulación en tres dimensiones, aplicando este principio al sistema GPS.

### 2.3.2 Triangulación Aplicada En El GPS

En el sistema NAVSTAR-GPS, la posición del receptor GPS (del usuario) es obtenida a partir de las coordenadas de los satélites de la constelación GPS (puntos de referencia conocidos) y las distancias medidas desde ellos al receptor. A partir de esa información, el receptor aplica el concepto de triangulación y así puede encontrar su posición. En general se necesitan cuatro puntos de referencia para determinar unívocamente una posición en el espacio de tres dimensiones. Sin embargo, en el caso del sistema GPS se sabe que la posición del receptor debe encontrarse en algún punto de la superficie terrestre. Por lo tanto, esta superficie puede considerarse como una de las cuatro esferas que se usan en la determinación de la posición y sólo se necesitan otros tres puntos de referencia. La figura 2.5 ilustra esto.

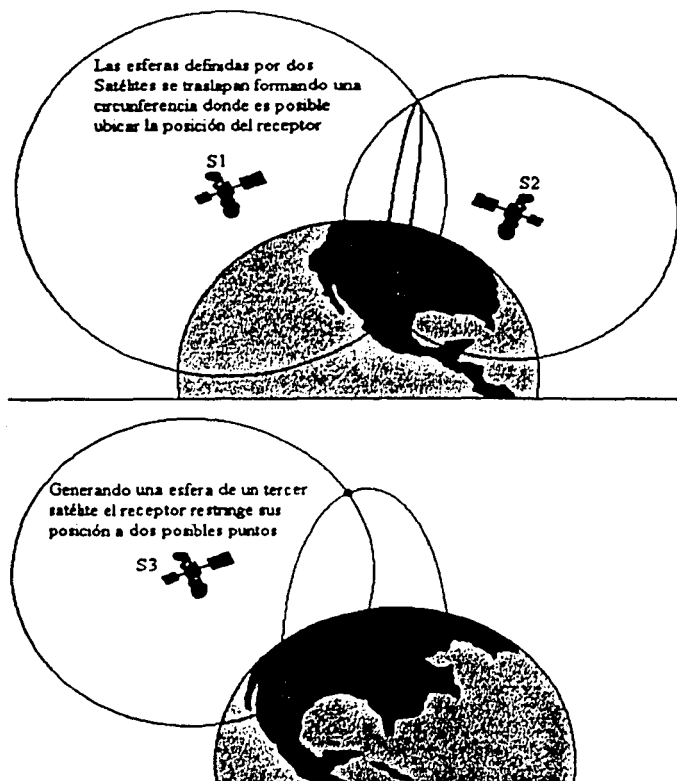


Figura 2.5 Determinación de la Posición del Receptor GPS.

Por medio de los satélites S1, S2 y S3 en la figura 2.5 la posición del receptor GPS puede restringirse a dos posibles puntos. Sólo uno de estos puntos se encuentra sobre la tierra, el otro punto está en el espacio. Matemáticamente, el receptor GPS elimina la posibilidad absurda de posicionamiento y localiza al receptor.

Lo anterior, sin embargo, sólo es posible si las distancias desde cada satélite al receptor han sido medidas con gran precisión. Como el receptor GPS no tiene una ecuación que se adapte exactamente a la superficie terrestre, debe estimar la altitud a la que se encuentra. Inicialmente el receptor considera su altitud como cero (en el centro de la tierra) e iterativamente incrementa esta altitud hasta que coincide con alguno de los dos posibles puntos de su posición. Si las distancias medidas a los satélites carecen de precisión, al receptor le será imposible estimar su altitud y a la vez el error en las distancias medidas. En tal caso, el receptor sólo calcula los valores de latitud y longitud de su posición, y no obtiene información de altitud. Si hay 4 o más satélites visibles por el receptor, se calculan las pseudo-distancias respecto a todos esos satélites y se obtiene un sistema con igual o mayor número de ecuaciones que incógnitas. Esto mejora el cálculo de la posición del receptor, pues se tienen más datos para calcular la corrección que debe hacerse a causa del error de medición y la altitud del receptor sobre la tierra.

Las distancias entre el receptor y los satélites se obtienen por medio de un 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 (es una onda electromagnética y por 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 a cada satélite es determinada haciendo uso de la fórmula  $d = ct$ , donde  $c$  corresponde a la velocidad de la luz en el vacío y  $t$  es el tiempo de recorrido de la señal desde el satélite hasta el receptor.

Para calcular la posición de un receptor, el GPS realiza el siguiente proceso:

- 1) Un satélite inicia su transmisión con un patrón digital como parte de su señal en ciertos intervalos de tiempo. Inicia la transmisión en un tiempo conocido, por ejemplo, exactamente a medianoche.
- 2) El receptor GPS inicia ejecutando el mismo patrón digital que el satélite, también a medianoche. Cuando la señal del satélite alcanza al receptor, su transmisión del patrón digital de referencia llegará unos bits detrás del patrón ejecutado por el receptor, pues ha pasado algo de tiempo 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 qué tan lejos ha viajado la señal. Si la señal ha viajado en línea recta, esta distancia sería la distancia entre el receptor y el satélite.
- 3) Cada satélite emite dos portadoras a la misma frecuencia. Estas portadoras están moduladas en fase (BPSK) por diferentes códigos pseudoaleatorios. El receptor GPS calcula la correlación entre el código recibido y el código de los satélites cuyas señales pretende detectar. De esta forma puede separar las señales de los diferentes satélites y obtener el retardo temporal de cada uno de ellos. Mediante el cálculo de la distancia se obtiene la ecuación de una esfera por cada satélite presente. La intersección de todas estas esferas dará la posición del receptor.

Evidentemente, es necesario proveer al sistema con un mecanismo para medir tiempo. Medir el tiempo sería sencillo si se supiera exactamente en qué momento las señales abandonan los satélites y en qué momento llegan al receptor, y resolver este asunto es el problema principal para los Sistemas de Posicionamiento Global. Una forma de resolverlo sería colocar relojes extremadamente precisos y en perfecta sincronía tanto en los satélites como en los receptores.

Debido a que no puede tenerse un reloj perfecto, los relojes en los receptores y en los satélites poseen un error que afectará la distancia medida. El sistema GPS procura que el error de los satélites sea mínimo utilizando relojes atómicos. Empero, para que todos los relojes estuvieran sincronizados se necesitarían relojes atómicos no solamente en los satélites, sino también en los receptores mismos (esto es así porque los tiempos que se miden para los cálculos de posición corresponden a escalas de nanosegundos (ns)), si bien los satélites cumplen esta condición, este no es el caso de los receptores, ya que su precio sería demasiado alto para el usuario.

El Sistema de Posicionamiento Global tiene una solución muy efectiva para este problema (un receptor GPS no posee un reloj atómico en absoluto, sino que posee un reloj de cuarzo, que es preciso, pero no lo suficiente). El receptor analiza todas las señales que recibe de los satélites y utiliza cálculos matemáticos para encontrar su posición y el tiempo exacto que viajó cada señal simultáneamente. Al hacer esto, introduce una variable más por determinar: el error de tiempo en el reloj del receptor, denominado *but (Bias clock User error in Time)*.

Así, la solución al problema es la introducción de esta nueva incógnita al sistema (además de las tres coordenadas espaciales del receptor) debida a la deriva que existe entre el reloj de los satélites y el reloj del receptor. En total, el receptor debe calcular cuatro variables: latitud, longitud, altitud y el error de tiempo para el reloj del receptor GPS.

Puesto que el receptor realiza todos los cálculos de tiempo (y por tanto de distancia), utilizando el reloj que posee internamente, todas las distancias serán proporcionalmente incorrectas. El receptor puede calcular el ajuste de distancia que permitirá que las cuatro esferas con centro en la posición de los satélites puedan intersectarse en un punto que pertenece a la superficie terrestre (estrictamente, a la superficie o a la atmósfera terrestre) ajustando el valor de la altitud. Entonces, para fijar una posición lo suficientemente precisa se necesitan 4 satélites como mínimo y no 3 como la teoría lo indica por principio. De hecho, si los relojes del receptor GPS y de los satélites se encuentran perfectamente sincronizados, la posición calculada respecto a cualesquiera tres satélites de cuatro a la vista debería ser exactamente la misma.

Debido a que el intervalo de tiempo es calculado a partir de relojes imperfectos cuyos errores deben corregirse, se usa el término de 'pseudo-distancias' para hacer referencia a estas distancias medidas.

Si la distancia a cada satélite desde el receptor se calcula teóricamente como:

$$D = \Delta T c \quad \text{Donde } \Delta T \text{ es el retraso temporal y } c \text{ es la velocidad de la luz.}$$

Se nota que en este cálculo intervienen dos cantidades: una cantidad definida  $c$  que ha sido medida previamente con gran precisión ( $c = 2.99792458 \times 10^8$  (m/s)); y la cantidad  $\Delta T$ , que se mide en el receptor a través de un procesamiento de la señal recibida del satélite. Así, debe considerarse que el valor de  $\Delta T$  tiene un error de medición asociado.

De este modo:

$D' = \Delta T_m c$  Es la pseudo-distancia calculada a cada satélite.  $\Delta T_m$ , el retraso temporal *medido*.

$\Delta T_m = \Delta T + \tau$  Donde  $\Delta T$  es el retardo temporal ideal y  $\tau$  representa el error de medición en unidades de tiempo.

Finalmente, pueden establecerse las siguientes relaciones:

$D' = D + c\tau$  La pseudo-distancia es igual a la distancia teórica más un error de medición.

$c\tau = c bu = bu$  El error de medición en distancia es prácticamente igual al error temporal del reloj del receptor multiplicado por la velocidad de la luz.  $bu$  es el error de reloj del receptor en unidades de distancia.

Éste es el error producido como consecuencia de la deriva existente entre el reloj de cada satélite y el reloj del receptor. Además de los errores de reloj, debe considerarse que en la medición de pseudo-distancias la señal proveniente de los satélites cambiará su velocidad de propagación al atravesar capas atmosféricas de distinta densidad, lo que introduce otro error en la posición. También, debe recordarse que la posición de observación es determinada a partir de las coordenadas de los satélites. La distancia medida, por lo tanto, también se encuentra afectada por las distintas perturbaciones orbitales que sacan a los satélites de sus órbitas teóricas. A pesar de estos diversos tipos de error, es el error de reloj el que más afecta la determinación de las distancias a los satélites.

En resumen: la función esencial de un receptor GPS es capturar la transmisión de al menos cuatro satélites y combinar la información presente en esas transmisiones con información de un almanaque electrónico, así puede determinar matemáticamente su posición en la Tierra. La información básica que provee un receptor es latitud, longitud y altitud (o un tipo de medida similar) de su posición actual. El sistema GPS, además de la posición, ofrece una referencia temporal muy exacta, esto permite sincronizar los relojes locales. Si cuenta con más de cuatro satélites, el receptor también puede reportar la velocidad a la que se mueve. Algunos receptores combinan estos datos con otra información, como mapas, para convertirse en un aparato más amigable con el usuario. Se pueden utilizar los mapas almacenados en la memoria del receptor, conectarlo a una computadora que posea mapas más detallados o simplemente conseguir un mapa del área y utilizar el receptor para trazar su ubicación.

## 2.4 CARACTERÍSTICAS DE LA SEÑAL GPS

Todas las transmisiones satelitales GPS se derivan de una frecuencia base  $f_0$  de 10.23 (MHz) que se obtiene del conjunto de relojes atómicos que van a bordo de cada satélite. Los satélites transmiten a frecuencias  $L1=1575.42$  (MHz) y  $L2=1227.6$  (MHz) moduladas con dos tipos de código y un mensaje de navegación. Las dos señales portadoras en la banda L ( $L1$  y  $L2$ ) se generan mediante la multiplicación entera de  $f_0$  de la siguiente manera:

$$f_0 = 10.23 \text{ (MHz)} \quad \text{Portadora } L1 = 154f_0 = 1575.42 \text{ (MHz), } \lambda_1 = 19 \text{ (cm)}$$

$$\text{Portadora } L2 = 120f_0 = 1227.60 \text{ (MHz), } \lambda_2 = 24.4 \text{ (cm)}$$

Para lograr obtener las lecturas de los satélites, se hace uso de dos códigos. Estos códigos se caracterizan por contener en ellos un ruido pseudo aleatorio (*Pseudo Random Noise*). El primer código es llamado código C/A (*Coarse/Adquisition, Común/Adquisición*) que se repite cada milisegundo y es generado con una frecuencia igual a  $f_0/10$ . El segundo código es llamado código P (Código Preciso) generado mediante una frecuencia igual a  $f_0$ . El código P se repite aproximadamente cada 266.4 días[5]. Las señales portadoras L1 y L2 son moduladas con el código P mientras que el código C/A se emplea para modular L1 solamente. Como una opción controlada desde tierra, el código C/A puede ser empleado también sobre L2.

| Código     | Frecuencia                   | Portadora |
|------------|------------------------------|-----------|
| Código P   | $f_0 = 10.23 \text{ MHz}$    | L1 y L2   |
| Código C/A | $f_0/10 = 1.023 \text{ MHz}$ | L1        |

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

El 31 de Enero de 1994, el tan esperado *antispoofing* (AS) fue permanentemente implementado. El propósito del AS es lograr que el código P esté disponible sólo para usuarios autorizados. Los usuarios deberán estar equipados con un dispositivo descryptador o la '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 es la base del Servicio de Posicionamiento Estándar (SPS por sus siglas en inglés). Este código 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. Es importante resaltar que el código C/A es completamente accesible para los civiles.

Debido a que los tiempos de 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, es 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.

## 2.5 EL RECEPTOR GPS

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:

### Por sus 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 militares.
- Receptores civiles.
- Receptores para navegación.
- Receptores geodésicos.

### 2.5.1 Elementos Esenciales De Un Receptor GPS

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* por sus siglas en inglés); una sección de rastreo y correlación de señal; y un microprocesador que controle el receptor, procese las señales y compute las coordenadas del receptor. El receptor también incluye una fuente de poder y un dispositivo de memoria para almacenar instrucciones y datos. La figura 2.6 muestra el diagrama de bloques de un receptor GPS donde se observan estos elementos esenciales.

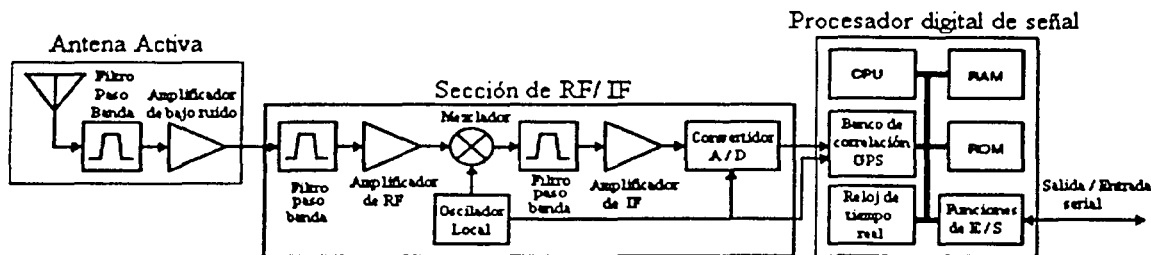


Figura 2.6 Diagrama de Bloques de un Receptor GPS

**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 débiles. Este tipo de unidad se utiliza cuando la antena se encuentra separada cierta distancia del receptor.

**Sección de RF/IF.-** La sección de RF/IF del receptor se utiliza para trasladar la frecuencia de la señal que llega a la antena hacia una frecuencia más baja, llamada frecuencia intermedia o IF, la cual es más fácil de manejar por los componentes del receptor. Un oscilador local realiza esta tarea combinando la



señal RF con una señal senoidal pura o tono. La mayoría de las unidades GPS utilizan osciladores de cristal de cuarzo de precisión, versiones mejoradas de aquellos encontrados en los relojes de pulso.

Algunos equipos de calidad geodésica proveen la opción de obtener la señal del oscilador local de una fuente externa tal como una frecuencia estándar atómica (de vapor de rubidio, un haz de cesio o de hidrógeno) que tienen una mayor estabilidad en la frecuencia. Los receptores dotados de tal señal pueden producir mediciones de fase de portadora con menos error de reloj.

La señal IF contiene toda la modulación presente en la señal transmitida, sólo que la portadora ha sido desplazada en frecuencia. La frecuencia de portadora desplazada es simplemente la diferencia entre la frecuencia de portadora original recibida y aquella del oscilador local. Algunos receptores emplean múltiples estadios de frecuencias intermedias, reduciendo la frecuencia de portadora a través de varios pasos. Se usan filtros en cada estadio de frecuencia intermedia para suprimir la interferencia fuera de banda y otras señales no deseadas. La señal final de IF pasa a la sección de rastreo y correlación del receptor. En la actualidad, se han desarrollado receptores GPS que no presentan la sección de RF/IF, pues son capaces de procesar la señal GPS en su frecuencia de transmisión (1.5 GHz) esto disminuye las dimensiones del receptor y el error introducido por los osciladores de los receptores clásicos.

**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-rango 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 L1 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 o SBAS (Sistema de Aumento Basado en Satélite), las señales en la frecuencia L2 para corrección de retraso debido a la ionosfera, o ambos.

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 multiplexación 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 los pseudo-rangos 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, y por un proceso de correlación cruzada. 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 (*DSP* por sus siglas en inglés) estos son circuitos integrados muy eficientes y pequeños.

### 2.5.2 Especificaciones Técnicas Del Receptor GPS Utilizado

El receptor GPS utilizado durante el desarrollo de este proyecto es el modelo FV-12 del fabricante *San José Navigation*. Es un receptor con 12 canales para código C/A, diseñado para navegación y de alto desempeño. Ofrece una salida de temporización en tiempo real y una salida de acuerdo a la norma NMEA 0183. Posee entrada para DGPS y una memoria integrada no volátil. Además del módulo receptor, se usó una miniantena para rastreo GPS del mismo fabricante. A continuación se presentan las características técnicas de cada uno de estos componentes.

#### Antena MK-4 (*Mini GPS Antenna for Micro Tracking*)

La antena MK-4 es la antena más pequeña que se encuentra en el mercado. Su tecnología de punta conjunta un tamaño pequeño con un buen desempeño. Con buena cobertura en casi todo el horizonte, trabaja bien en terrenos con denso follaje o en ciudades, así como en medios donde existe interferencia electromagnética. La antena MK-4 es compatible con la mayoría de los modelos de receptores GPS y es una excelente alternativa para muchas aplicaciones del GPS incluyendo navegación vehicular, aviación y en la milicia. La figura 2.7 muestra el diagrama esquemático de la antena MK-4.

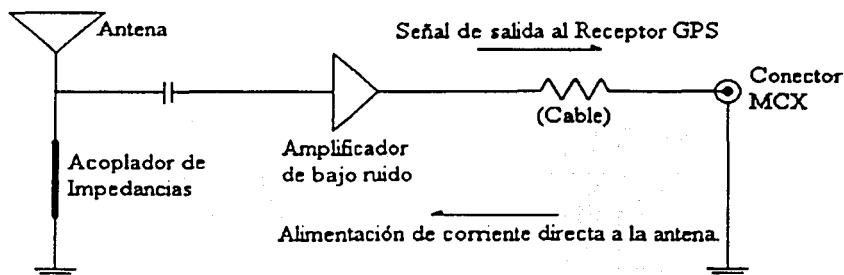


Figura 2.7 Diagrama Esquemático de la Antena GPS

La tabla 2.2 condensa las especificaciones de la Antena, el Amplificador de bajo ruido y el Cable en conjunto. La tabla 2.3, las especificaciones ambientales para su uso.

|                      |                 |                |           |
|----------------------|-----------------|----------------|-----------|
| Frecuencia central   | 1575.42 (MHz)   | Ancho de banda | 2 (MHz)   |
| Ganancia             | 24 dB min.      | Relación axial | 3 dB max. |
| Figura de ruido      | 2.0 max.        | VSWR           | 1.5 max.  |
| Impedancia de salida | 50 ( $\Omega$ ) |                |           |

Tabla 2.2 Especificaciones Técnicas de la Antena GPS

|                                   |                          |
|-----------------------------------|--------------------------|
| Rango de Temperatura de Operación | - 30° C ~+ 85° C         |
| Rango de Temperatura Establecida  | - 40° C ~+ 90° C         |
| Rango de Humedad Relativa         | 95 % no-condensación     |
| Resistencia al agua               | 100 % resistente al agua |

Tabla 2.3 Especificaciones Medioambientales Para el Uso de la Antena

La figura 2.8 muestra el diagrama técnico con las dimensiones de la antena GPS. Puede señalarse que se trata de un dispositivo bastante pequeño, lo que resulta muy conveniente al momento de su instalación. La cubierta de la antena es plástica mientras que su base es metálica e incluye un imán para que pueda ser colocada en el techo de un automóvil, barco, etc. La posición óptima de colocación de la antena es con su base horizontal, y debe mantenerse en espacio abierto para recibir correctamente la señal de los satélites GPS.

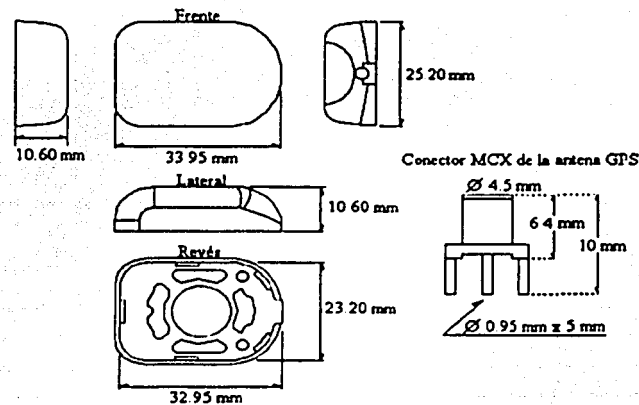


Figura 2.8 Diagrama Técnico de la Antena GPS

### Módulo Receptor GPS FV-12

Este módulo receptor es una de las maquinarias GPS más pequeñas del mundo para utilizarse en aplicaciones de diseño y evaluación. Incorpora el rastreo de 12 satélites con una excelente re-adquisición de la señal en momentos de obstrucción. Además de la salida estándar NMEA-0183, ofrece la salida de datos binaria SiRF para aplicaciones avanzadas o de mayor precisión. Algunas de sus funciones de operación son:

- Renueva la posición de los satélites en una décima de segundo después de salir de una región bloqueada. Esto es crítico para restablecer la precisión de la posición mientras que un móvil atraviesa puentes dentro de las ciudades o zonas de denso follaje.
- Cuenta con un sistema de habilitación del GPS para proveer información de la posición durante intervalos en los que sólo un satélite se encuentra visible.
- Elimina efectivamente la información que proviene de señales que no hayan tomado una trayectoria directa.

| General  | Precisión                                     |
|--|---|
| Frecuencia: 1575.42 (MHz)                                    | Posición: 25 (m) CEP sin SA                   |
| Código C/A : 1.023 (MHz) de velocidad de bits                | Velocidad: 0.1 (m/s) sin SA                   |
| Canales: 12 canales  | Tiempo: 1 (µs) sincronizado con el tiempo GPS |
| Tasa de adquisición de datos                                 | Condiciones dinámicas                         |
| Readquisición: 0.1 (s) en promedio                           | Altitud: 18000 (m) máximo                     |
| Encendido caliente: 8 (s) en promedio                        | Velocidad: 515 (m/s) máximo                   |
| Encendido tibio: 38 (s) en promedio                          | Aceleración: 4g máximo                        |
| Encendido frío: 48 (s) en promedio                           |   |
| Potencia   | Tiempo – pulsos de 1-PPS                      |
| Potencia principal: 5 (Vdc) +/- 5%                           | Nivel: TTL                                    |
| Fuente de corriente: 180 (mA) típico                         | Duración de pulso: 100 (ms)                   |
| Potencia de respaldo: 2.5 (V) ~ 4.5 (V)                      | Referencia: En el flanco negativo del pulso.  |
| Corriente de respaldo: 10 (µA) típico                        | Mediciones: Alineadas a 1 (s) GPS, +/- 1 (µs) |
| Comunicación   | Ambiente                                      |
| Interface: 2 comunicaciones seriales full duplex.            | Temperatura de operación: -40°C ~ +85°C       |
| Protocolo de mensaje: SiRF & NMEA0183                        | Temperatura de almacenaje: -55°C ~ +100°C     |
| SiRF binario: posición, velocidad, altitud, estado & control |   |
| NMEA0183: CGA, GLL, GSA, GSV, RMC & VTG                      |   |
| Salida por default: Todos los mensajes NMEA, baudaje 4800    |   |

Tabla 2.4 Especificaciones Técnicas del Módulo Receptor GPS FV-12

|              |                          |              |  |
|--------------|--------------------------|--------------|--|
| VANT 01      | <input type="checkbox"/> | 02 VDC       | VANT: Voltaje de alimentación para antena          |
| BAT 03       | <input type="checkbox"/> | 04 VDC       | VDC + 3.8(V) a 6.5 (V) de alimentación             |
| PBRES 05     | <input type="checkbox"/> | 06 RESERVADO | BAT: Voltaje de batería de respaldo                |
| SELECT 07    | <input type="checkbox"/> | 08 RESERVADO | PBRES: Entrada de Reinicio (RESET) Activo en Bajo  |
| RESERVADO 09 | <input type="checkbox"/> | 10 GND       | SELECT: Carga de Datos de RS-232 a ROM             |
| TXA 11       | <input type="checkbox"/> | 12 RXA       | GND: Tierra  |
| GND 13       | <input type="checkbox"/> | 14 TXB       | TXA: Salida de Datos Señal A                       |
| RXB 15       | <input type="checkbox"/> | 16 GND       | RXA: Entrada de Datos Señal A                      |
| RESERVADO 17 | <input type="checkbox"/> | 18 GND       | TXB Salida de Datos Serial B                       |
| TIMEMARK 19  | <input type="checkbox"/> | 20 RESERVADO | RXB: Entrada de Datos Serial B                     |
|              |                          |              | TIMEMARK: Salida de Temporización, 1 pulso por seg |

Figura 2.9 Asignación de Pines del Receptor GPS

## 2.6 EL PROTOCOLO DE COMUNICACIÓN NMEA 0183

### 2.6.1 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 ocupaciones con equipo electrónico marítimo periférico. La Norma NMEA 0183 define una interfaz eléctrica y un protocolo de datos para la comunicación entre instrumentos marinos [6].

Los instrumentos en la norma NMEA 0183 son designados como 'parlantes' *talkers* o como 'oyentes' *listeners* (con algunos instrumentos que poseen ambas designaciones), utilizando una interfaz serial asincrónica con los parámetros que muestra la tabla 2.5. El NMEA 0183 es un estándar industrial voluntario, su primera publicación fue en Marzo de 1983. Se ha ido actualizando con el tiempo, la última publicación, versión 2.30, Marzo 1998, está disponible en las oficinas de la NMEA.

|                         |                |
|-------------------------|----------------|
| Baudaje                 | 4800           |
| Número de bits de datos | 8 (bit 7 es 0) |
| Bits de Parada          | 1(o más)       |
| Paridad                 | Ninguna        |
| Saludo                  | Ninguno        |

Tabla 2.5 Parámetros de la Interfaz Asíncrona NMEA 0183

### 2.6.2 Interfaz Eléctrica

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

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

En ambos casos, el circuito receptor recomendado utiliza un optoacoplador con circuitos de protección. La entrada debe aislarse de la tierra del receptor. En la práctica, el cable sencillo, o el cable A del RS-422 puede ser conectado directamente a la entrada RS-232 de una computadora. De hecho, muchos de los últimos productos, como los receptores GPS de mano, no tienen una salida diferencial RS-422, sólo tienen una línea sencilla con nivel de señal TTL o 5 (V) CMOS compatible.

### 2.6.3 Formato General De Los Mensajes

En la norma NMEA 0183 todos los datos son transmitidos como 'mensajes'. Únicamente se permiten caracteres ASCII desplegados en pantalla, más <CR> (retorno de carro, *carriage return*) y <LF> (línea nueva o *line feed*). Cada mensaje comienza con el signo "\$" y termina con <CR><LF>. Existen tres tipos básicos de mensajes: mensajes de parlante (*talker sentences*), mensajes de propietario (*proprietary sentences*) y mensajes de petición (*query sentences*). Para el sistema presentado los mensajes que interesan son los mensajes de parlante, estos son, los mensajes de navegación que mandan los satélites GPS. El formato general de los mensajes de parlante es:

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

Las primeras dos letras que siguen a "\$" forman el 'identificador de parlante', *talker identifier*. Los siguientes tres caracteres, representados por "sss", forman el 'identificador de mensaje' *sentence identifier*, y son seguidos por un número variable de campos de datos separados por comas, seguidos por un *checksum* opcional, y finalmente el mensaje termina con <CR><LF>. Los campos de datos son definidos de manera única para cada tipo de mensaje. Un 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 que sigue. Los caracteres "238" son el valor del encabezado, y "M" define el valor del encabezado como magnético.

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

### 2.6.4 Formato Del Mensaje GPS

El identificador de parlante o *talker* que se utiliza para el Sistema de Posicionamiento Global es "GP". Los satélites GPS son capaces de transmitir varios tipos de mensajes, cada uno de los cuales contiene diferente información. Como ejemplos tenemos el mensaje VTG, que reporta el curso y la velocidad en tierra del receptor; el mensaje GLL que contiene la posición geográfica (latitud y longitud) del receptor; el mensaje GSA que informa el número de satélites activos y la dilución de precisión; etc. Este proyecto utilizó únicamente el mensaje GGA que es el mensaje diseñado para uso marino. El formato de este mensaje es el siguiente:

GGA.- Datos de ajuste del GPS. Tiempo, Posición y Datos de Ajuste Relacionados para un receptor GPS.

```
$GPGGA,hhmmss.ss,lll.ll,a,yyyyy.yy,a,x,xx,x.x,x.x,M,x.x,M,x.x,xxxx*hh
  1      2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

Los 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. Número de satélites a la vista (de 00 a 12)
8. Dilusión horizontal de precisión
9. Altitud de antena arriba / abajo del nivel medio del mar
10. Altitud en unidades de antena, metros
11. Separación geoidal
12. Unidades de separación geoidal
13. Datos de edad de GPS diferencial
14. Identificación de estación de referencia diferencial
15. Checksum

Los principales campos de interés para este proyecto son:

| No. de Campo & Información | No. de Caracteres | Formato de información * |
|----------------------------|-------------------|--------------------------|
| 2) Latitud                 | 7: IIII.II        | ggmm.dd                  |
| 3) N o S                   | 1: a              | -                        |
| 4) Longitud                | 8: yyyyy.yy       | gggmm.dd                 |
| 5) E ó W                   | 1: a              | -                        |

\* g = grados, m = minutos, d = décimas de minutos

La altitud del receptor GPS no interesa en este proyecto, pues se sabe que el móvil permanecerá a nivel del mar (o no cambiará considerablemente su altitud). En el capítulo seis de este escrito se describe el tratamiento que se da al mensaje GPGGA de acuerdo con las necesidades del proyecto. Asimismo, en el Capítulo 5 se presenta el algoritmo de navegación diseñado.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] Leick, Alfred; "GPS Satellite Surveying"; 2ª Edición, 1995; John Wiley & Sons, Inc.
- [2] Ibid. pp. 388 – 392.
- [3] Federal Aviation Administration; Sistema WAAS; <http://gps.faa.gov/Programs/WAAS/waas.htm>  
European Space Agency; Sistema EGNOS; <http://www.esa.int/export/esaSA/navigation.html>
- [4] Bao-Yen, James; "Fundamentals of Global Positioning System Receivers, A Software Approach"; pp. 8 –29; 2000; John Wiley & Sons, Inc.
- [5] Leick; 1995; pag. 72.
- [6] National Marine Electronic Association; "NMEA 0183 Protocol"; Versión 2.30; Enero 2000; [www.nmea.org](http://www.nmea.org)

## 3

## El Microprocesador M68HC11

### 3.1 EL MICROPROCESADOR

Los microprocesadores se empezaron a utilizar en algunos productos desplazando la lógica discreta, esto es, reemplazando los diseños lógicos convencionales que utilizaban flip-flops, compuertas, contadores, registros y otras funciones con integración a media escala (*MSI, Medium Scale Integration*). Por ejemplo, un semáforo que anteriormente utilizaba 200 circuitos TTL, ahora puede armarse con pocos circuitos integrados (*IC, integrated circuit*) que utilizan sistemas basados en  $\mu$ Ps.

Algunas ventajas de la superioridad de los  $\mu$ Ps en comparación con diseños lógicos discretos son:

- Disminución de circuitos integrados, tarjetas impresas y conectores, que resulta en una disminución en el costo de ensamblaje.
- Mayor confianza en el diseño, debido a la disminución de interconexiones entre circuitos integrados.
- Menores requerimientos de potencia, lo que permite que el diseño de las fuentes de poder sea más sencillo.
- Simplificación en la evaluación, prueba y rediseño del sistema. Debido a que los equipos basados en  $\mu$ Ps operan bajo el control de un programa guardado en memoria, su operación es fácilmente modificable, basta con cambiar el programa o reemplazar la memoria en vez de cambiar el cableado en un sistema con lógica discreta.

Todas las microcomputadoras y los microcontroladores, a pesar de variar en su arquitectura, tienen en común el microprocesador  $\mu$ P, el elemento principal, usualmente en un solo chip de alta escala de integración que contiene la unidad de control y la Unidad Lógico Aritmética (ULA). Es común referirse al microprocesador como *MPU (Microprocessor Unit)*, que es el corazón de la  $\mu$ C, porque sus capacidades determinan las capacidades propias de la  $\mu$ C. La velocidad del MPU limita la velocidad máxima; las direcciones y *pin*s del MPU determinan la capacidad de memoria y el tamaño de la palabra de la  $\mu$ C; y sus *pin*s de control determinan el tipo de interfaces de entrada / salida requeridas.

Algunas las funciones que realiza el MPU son: proveer señales de control y señales de tiempo a todos los elementos de la  $\mu$ C; cargar los datos y comandos de la memoria; transferir los datos de / a los instrumentos de entrada / salida; decodificar las instrucciones; realizar operaciones aritméticas y lógicas requeridas por sus instrucciones; responder a las señales de control de entrada / salida como *RESET* e *INTERRUPT*, etc. El MPU contiene toda la circuitería lógica para desarrollar estas funciones, pero esta lógica interna del MPU no es accesible externamente. La circuitería interna del MPU sólo puede modificarse mediante el programa que se guarda en su memoria, para que el MPU lo ejecute. Por ello es que el MPU es tan flexible y versátil.



En una aplicación común, la microcomputadora ( $\mu\text{C}$ ) recibe información o datos de entrada, ejecuta una secuencia de instrucciones que de alguna manera manipulan o procesan esta información y luego entrega información o datos de salida. En las operaciones de entrada y salida intervienen las interfaces con el entorno de la microcomputadora, mientras que las funciones de procesamiento se realizan dentro de la  $\mu\text{C}$  bajo el control de un programa establecido en la memoria (RAM ó ROM).

Por ejemplo, en un sistema de pantalla que despliega el estado del motor de un auto y que está controlado por una  $\mu\text{C}$ , la información de entrada consiste de algunos parámetros del motor tales como la temperatura, la presión del aceite y la tasa de consumo de combustible. Cada una de estas cantidades físicas es convertida en una cantidad eléctrica a través de un transductor. Las cantidades eléctricas son convertidas en una señal digital para ser introducidas a la  $\mu\text{C}$ . La  $\mu\text{C}$  procesa los datos de entrada (los coloca en el formato correcto para ser desplegados) y finalmente los despliega en pantalla para que el usuario pueda leerlos.

En el sistema de navegación que se presenta en este documento, las señales de entrada a la  $\mu\text{C}$  provienen del receptor GPS, del sistema de sensores y de un teclado mediante el cual se pueden especificar las coordenadas de destino para el móvil. La  $\mu\text{C}$  procesa estas entradas para generar un conjunto de señales de control para los motores del vehículo y para el sistema de sensores. Una señal de salida durante el diseño del sistema fue también cierta información de posicionamiento del móvil desplegable en la pantalla de una Computadora Personal (PC). Como puede verse, a pesar de la particularidad de los dispositivos utilizados, se trata de una aplicación típica para una  $\mu\text{C}$ .

### **3.2 ESPECIFICACIONES TÉCNICAS DEL MICROPROCESADOR M68HC11**

En el desarrollo de este proyecto se utilizó el microprocesador M68HC11E9. La serie E de los microprocesadores M68HC11 consiste de microcontroladores de 8 bits (este es el tamaño de los registros o conjunto de datos que el MPU es capaz de manejar a un tiempo, y almacenar en una localidad de memoria) que combinan la unidad central de procesamiento con dispositivos periféricos de alto rendimiento integrados en un solo chip.

Dentro de la serie E existen muchas presentaciones de microcontroladores que varían según sus configuraciones de memoria: de acceso aleatorio (RAM), de sólo lectura (ROM), programable y borrable de sólo lectura (EPROM), programable borrable eléctricamente de sólo lectura (EEPROM), etc. Con excepción de algunas pequeñas diferencias, todos los microcontroladores de la serie E funcionan de la misma manera. Todos operan a frecuencias desde 3 MHz a DC con poco consumo de potencia.

Los microprocesadores de la serie E tienen la estructura que muestra la figura 3.1, en particular, en este proyecto se utilizó la configuración que presenta la figura 3.2. La potencia para el dispositivo se administra a través de VDD y VSS (VCC y GND). VDD es un voltaje positivo respecto a VSS, que es tierra lógica. EL MCU opera desde una fuente de alimentación nominal de 5 (V).

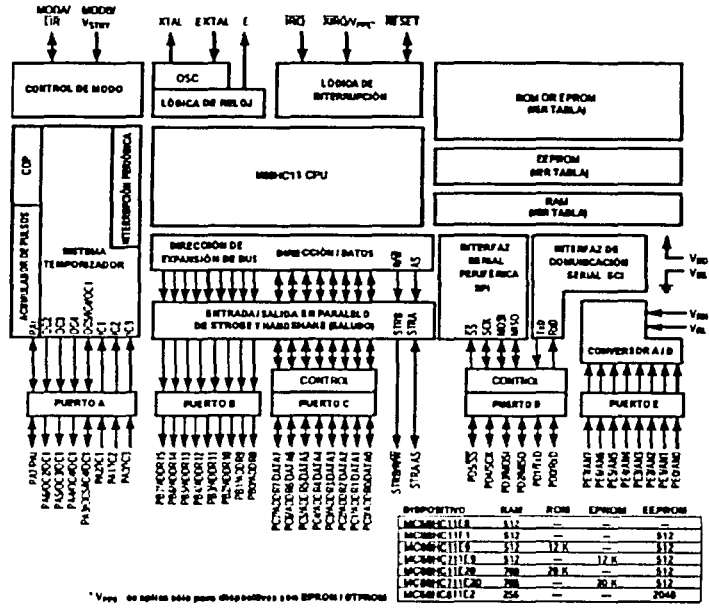
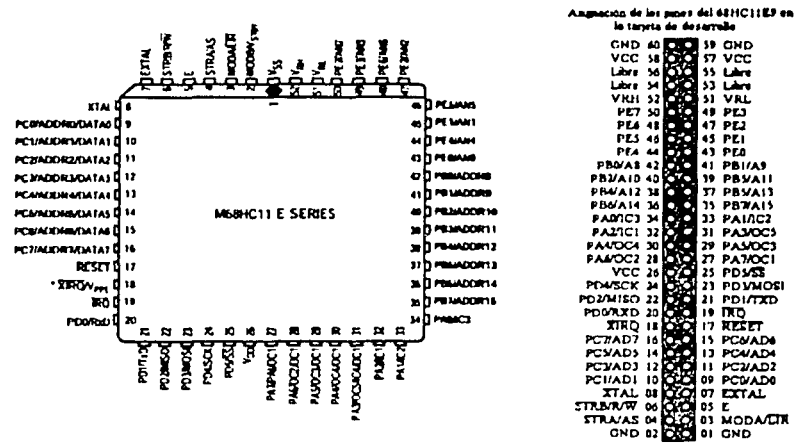


Figura 3.1 Diagrama de Bloques del Microprocesador 68HC11E9



Asignación de los pines del 68HC11E9 en la tarjeta de desarrollo

|          |    |    |         |
|----------|----|----|---------|
| GND      | 40 | 59 | GND     |
| VCC      | 58 | 57 | VCC     |
| Libre    | 56 | 55 | Libre   |
| Libre    | 54 | 53 | Libre   |
| VRL      | 52 | 51 | VRL     |
| PET      | 50 | 49 | PE3     |
| PE4      | 48 | 47 | PE2     |
| PE5      | 46 | 45 | PE1     |
| PE4      | 44 | 43 | PE0     |
| PBWA8    | 42 | 41 | PBWA8   |
| PBWA10   | 40 | 39 | PBWA11  |
| PBWA12   | 38 | 37 | PBWA13  |
| PBWA14   | 36 | 35 | PBWA15  |
| PAWIC3   | 34 | 33 | PAWIC2  |
| PAWIC1   | 32 | 31 | PAWOC3  |
| PAWOC4   | 30 | 29 | PAWOC1  |
| PAWOC2   | 28 | 27 | PAWOC1  |
| VCC      | 26 | 25 | PDVSS   |
| PDWVSC   | 24 | 23 | PDVMO51 |
| PDWVMO   | 22 | 21 | PDWVMO  |
| PDWVMO   | 20 | 19 | TRD     |
| XTAL     | 18 | 17 | RESET   |
| PC7AD7   | 16 | 15 | PC7AD0  |
| PC7AD5   | 14 | 13 | PC7AD1  |
| PC7AD3   | 12 | 11 | PC7AD2  |
| PC7AD1   | 10 | 09 | PC7AD0  |
| XTAL     | 08 | 07 | EXTAL   |
| STRB/R/W | 06 | 05 | E       |
| STRB/R/W | 04 | 03 | MODA/DT |
| GND      | 02 | 01 | GND     |

Figura 3.2 Asignación de Pines para la Configuración de 52 pines

### 3.3 LOS PUERTOS Y REGISTROS DEL MICROPROCESADOR

#### 3.3.1 Puertos Del Microprocesador

**Puerto A.-** En todos los modos de operación, el puerto A se puede configurar para tener tres funciones de entrada de captura de tiempo (IC) y cuatro funciones para salida de comparación (OC). Un *pin* adicional puede configurarse como salida o entrada de comparación

**Puerto B.-** Durante el modo de operación de chip único, todos los *pines* del Puerto B son salidas de propósito general. El Puerto B también puede ser utilizado en el modo de salida de *strobe* simple. En este modo, un pulso de salida aparece en la señal STRB cada vez que un dato se escribe en el Puerto.

**Puerto C.-** En el modo de operación de chip único, todos los *pines* del Puerto C son *pines* de entrada / salida de propósito general. El Puerto C también puede usarse en modo de saludo (*handshake*) o en modo de Entrada / Salida en paralelo donde la señal de entrada STRA y la de salida STRB actúan como líneas de control de saludo.

**Puerto D.-** Los *pines* PD0-PD5 ubicados en el puerto D pueden ser utilizados para señales de entrada / salida de propósito general. Esto *pines* sirven alternadamente como una interfaz de comunicación serial (SCI) y como señales de interfaz serial periféricas (SPI) cuando esos subsistemas se habilitan.

**Puerto E.-** Los *pines* del puerto E se utilizan como una entrada de 8 bits de propósito general o en conjunción con el convertidor analógico-digital (A/D). Cuando se usa como parte del sistema convertidor A/D, permite realizar conversiones sin necesidad de una interfaz externa. El sistema convertidor A/D del microprocesador se compone del puerto E, el convertidor A/D, el registro de control OPTION y el registro de control/estado A/D (ADCTL).

#### 3.3.2 Registros Del Microprocesador

Los registros del CPU M68HC11 son parte integral del CPU y no están direccionados como si fueran localidades de memoria. La figura 3.3 muestra estos registros en la forma del modelo de programación. Se conoce así a esta configuración debido a que contiene los registros básicos que intervienen en cualquier programa para el microprocesador. A partir de este modelo de programación puede representarse el movimiento de la mayor parte de los datos que intervienen en un proceso; las operaciones lógicas y matemáticas que se realizan sobre estos datos; y el monitoreo y control de ejecución del programa.

**Acumuladores A, B y D.-** Son registros de 8 bits de propósito general que guardan operandos y resultados de cálculos aritméticos o manipulaciones de datos. Para algunas instrucciones, estos dos acumuladores se utilizan como un acumulador de dos bytes (16 bits) llamado acumulador D. Por lo general los acumuladores A y B son intercambiables entre sí para el uso de instrucciones, existen sin embargo algunas excepciones.

**Registros Indexados X y Y.-** El registro IX provee un valor indexado de 16 bits que puede sumarse a una cantidad de ajuste de 8 bits dada por alguna instrucción para crear una dirección efectiva. El registro IX también puede usarse como un contador o como un registro de almacenamiento temporal. El registro IY realiza una función en modo indexado similar a la del registro IX. Sin embargo, la mayoría de las

funciones que utilizan el registro *IY* requieren de un byte extra en el código de máquina y un ciclo extra en el tiempo de ejecución debido a la manera en que se implementa el código hexadecimal.

**El Apuntador de Pila (*Stack Pointer SP*).**- EL CPU 68HC11 tiene una pila de programa automática (una pila es un conjunto de registros que ocupan localidades contiguas en memoria). Esta pila puede localizarse en cualquier parte del espacio de memoria y puede ser de cualquier tamaño dependiendo de la capacidad de memoria disponible en el sistema. Normalmente el *Stack Pointer* (*SP*) se inicializa por alguna de las primeras instrucciones en un programa de aplicación. Cada vez que un nuevo byte se coloca en la pila, el *SP* es decrementado y cada vez que un byte se extrae de la pila el *SP* se incrementa. En cualquier momento, el *SP* mantiene la dirección de 16 bits que corresponde a la siguiente locación de memoria que está libre en la pila.

**El Contador de Programa (*Program Counter PC*).**- El contador de programa es un registro de 16 bits que contiene la dirección de la siguiente instrucción que será ejecutada. Después de un *reset* o reinicio, el contador de programa se inicializa desde uno de seis posibles vectores, dependiendo del modo de operación y la causa del reinicio.

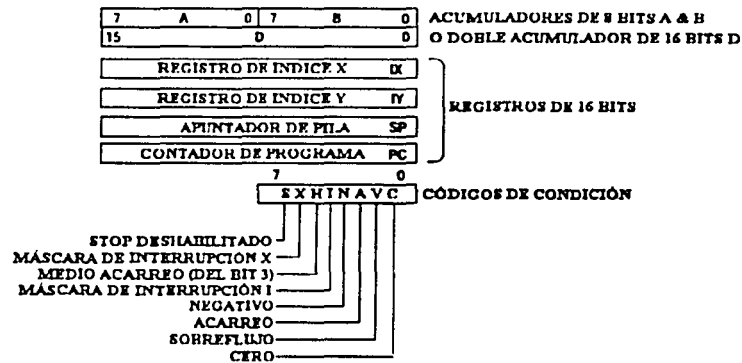


Figura 3.3 Modelo de Programación del Microprocesador M68HC11

**El Registro de Código de Condición (*CCR*).**- Contiene cinco indicadores (C, V, Z, N y H), dos bits de enmascaramiento de interrupción (*IRQ'* y *XIRQ'*) y un bit de deshabilitación de paro o *stop* (*S*).

- **Carry / Borrow (Acarreo) (C).**- Este bit se enciende si la unidad lógica aritmética (ALU) realiza un acarreo o pide prestado durante una operación aritmética..
- **Overflow (Sobreflujo) (V).**- Este bit se enciende si alguna operación causa un sobreflujo aritmético; de lo contrario se apaga.
- **Zero (Cero) (Z).**- Este bit se enciende si el resultado de alguna operación, ya sea aritmética, lógica o alguna manipulación de datos, es cero; de lo contrario el bit Z se apaga.
- **Negative (Negativo) (N).**- Este bit se enciende si el resultado de alguna operación, ya sea aritmética, lógica o de manipulación de datos, es negativo, o sea que el bit más significativo sea uno; de lo contrario el bit N se apaga.
- **Half Carry (Medio Acarreo) (H).**- Este bit se enciende cuando ocurre un acarreo entre los bits 3 y 4 de la unidad lógica aritmética durante las instrucciones ADD, ABA, o ADC; de lo contrario el bit H se apaga..
- **Stop Disable (Deshabilitado de Alto) (S).**- Al encender este bit se previene la ejecución de la instrucción STOP, la cual coloca al MPU en modo de alto con baja potencia donde todos los relojes y el oscilador del MPU se detienen.

- **Interrupt Mask (Máscara de Interrupción) (I).**- La máscara (bit I) de petición de interrupción (*Interrupt Request, IRQ*) es una máscara global que deshabilita todas las fuentes de interrupción enmascarables. Mientras el bit I esté encendido, no puede ocurrir ninguna interrupción, aunque pueden mantenerse pendientes hasta que el bit I se apague.
- **X Interrupt Mask (Máscara de Interrupción X) (X).**- El bit de máscara XIRQ (X) deshabilita las interrupciones que vienen desde el *pin* XIRQ'.

### 3.4 INTERFAZ DE COMUNICACIÓN SERIAL (SCI)

La interfaz de comunicación serial (SCI) es un transmisor- receptor asíncrono universal (UART), y constituye uno de los dos subsistemas independientes de entrada-salida serial dentro de los microcontroladores M68HC11 de la serie E. Tiene un formato estándar de no-regreso-a-cero (*NRZ-level*) con un bit de inicio, ocho o nueve bits de datos, y un bit de alto. Existen varias tasas de baudaje. El receptor y transmisor SCI son independientes pero utilizan el mismo formato de datos y la misma velocidad. El formato de datos requiere de las siguientes condiciones:

- Una línea de espera en estado alto antes de la transmisión o recepción de un mensaje.
- Un bit de inicio (cero lógico), transmitido o recibido, que indica el inicio de cada carácter.
- Que la información se transmita o se reciba con el bit menos significativo al inicio.
- Un bit de alto (uno lógico), utilizado para indicar el fin de la trama.
- Una trama consiste de un bit de inicio, un carácter de 8 ó 9 bits de datos, y un bit de alto.
- Un freno, definido como la transmisión o recepción de un cero lógico por un número múltiple de tramas.
- La selección del tamaño de la palabra es controlada por el bit M del registro de control SCCR1

**Transmisión.**- El transmisor SCI incluye un registro de datos de transmisión en paralelo (SCDR) y un registro de conmutación serial. El contenido del registro de conmutación serial se puede escribir sólo a través del SCDR. Esta operación de doble buffer permite que un carácter sea entregado de manera serial mientras que otro carácter espera en el SCDR para ser transferido al registro de conmutación serial. La salida del registro de conmutación serial se aplica a TxD siempre que la transmisión se encuentre en proceso o el bit de habilitación de transmisión (TE) del registro de control de comunicación serial 2 (SCCR2) esté encendido.

**Recepción.**- Durante las operaciones de recepción, la secuencia de transmisión es revertida. El registro de conmutación serial recibe el dato y lo transfiere al registro datos de recepción en paralelo (SCDR) como una palabra completa. Esta operación de doble buffer permite a un carácter ser tomado de manera serial mientras que otro carácter se encuentra en el SCDR.

### 3.5 MEMORIA DEL MICROPROCESADOR

El microprocesador cuenta con varios modos de operación: dos modos normales de operación llamados modo sencillo o de *single chip* y modo expandido; y dos modos de prueba llamados modo *bootstrap* y modo de *special test*. Dependiendo del modo de operación, se tienen diferentes configuraciones de memoria tal como muestra la figura 3.4. Debido al tamaño del programa escrito para este proyecto, fue necesario utilizar más memoria de la disponible en el microprocesador, por ello se requirió de una tarjeta de expansión que proporcionara memoria extra sin tener que perder un puerto del microprocesador para poder acceder a ella. La tabla 3.2 muestra el mapa de memoria con el uso de la tarjeta de expansión.

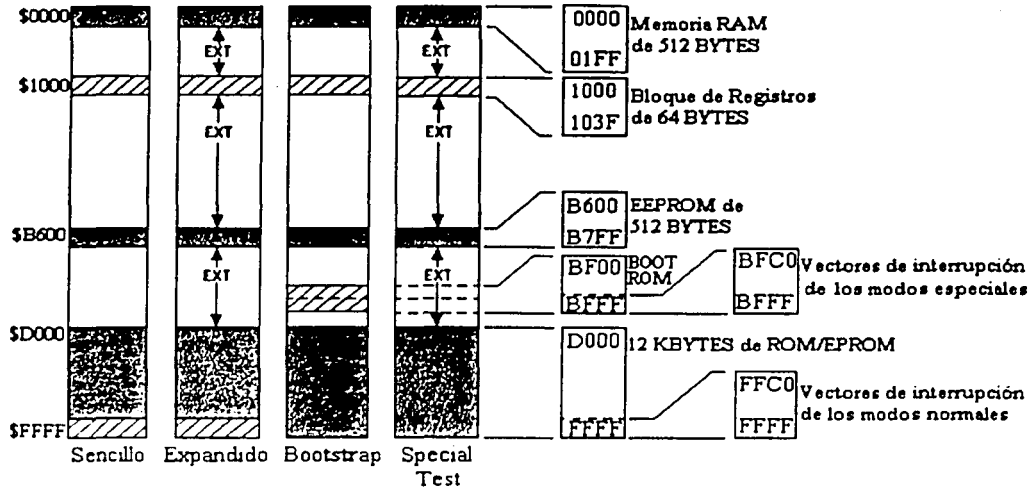


Figura 3.4 Mapa de Memoria del M68HC11 en Sus Distintos Modos de Operación

| Localidades de memoria | Asignación           |
|------------------------|----------------------|
| \$0000 a \$0FFF        | RAM                  |
| \$1000 a \$103F        | REGISTROS DE CONTROL |
| \$1040 a \$7FFF        | RAM                  |
| \$A800                 | COMANDOS DISPLAY     |
| \$B800                 | DATOS DISPLAY        |
| \$B600 a \$B7FF        | EEPROM               |
| \$E000 a \$HF          | BÚFALO               |

Tabla 3.2 Mapa de Memoria con Tarjeta de Expansión

### BIBLIOGRAFÍA GENERAL DEL CAPÍTULO 3

- Tocci, Ronald J. et.al.; "Microprocessors and Microcomputers"; Hardware and Software; 4ª edición, 1997; Prentice Hall.
- "M68HC11E Family"; Technical Data ;Motorola Inc.2000.

## 4

# Subsistemas

## 4.1 SENSORES

### 4.1.1 Teoría De Sensores

Los seres humanos tomamos por supuestos los asombrosos sistemas de percepción que poseemos. Por ejemplo, realizar la simple tarea de beber de una taza requiere una compleja interacción de percepción, interpretación, reconocimiento y coordinación, la cual ha sido entendida sólo mínimamente.

De este modo, alcanzar un desempeño del nivel humano en un robot se ha convertido en algo tremendamente difícil. Los juegos de computadora que vencen en ajedrez a los expertos son muy comunes hoy en día, pero un programa que reconoce, por ejemplo, una silla en un escenario arbitrario aún no existe. El equivalente a una computadora que se encuentra dentro de cada cabeza humana destina grandes cantidades de materia gris a los problemas de la percepción y la manipulación.

Mientras que a los diseñadores les gustaría que alguno de los robots que construyen entendiera y fuera consciente de su medio ambiente, en realidad un robot se encuentra muy limitado con el software y los sensores que sus creadores le dan. Los sensores son meramente transductores que convierten alguna cantidad física en señales eléctricas que el microprocesador del robot puede leer. Lo anterior puede realizarse utilizando un convertidor analógico-digital (A/D) vinculado al procesador; cargando un valor desde un puerto de entrada / salida (I/O por sus siglas en inglés); o usando una interrupción externa. Usualmente, es necesario que exista una interfaz entre el sensor y el microprocesador para acondicionar o amplificar la señal que interesa.

Con software, pueden crearse diferentes niveles de abstracción, para ayudar a los programadores a pensar en los datos recibidos por los sensores desde diferentes puntos de vista. En el nivel más alto, el sistema de inteligencia necesita manejar algunas variables: ¿Existe un obstáculo a la izquierda? ¿Hay un objeto que se mueve en la trayectoria del móvil? Sin embargo, las únicas preguntas que el robot es capaz de contestar, en su nivel más bajo de abstracción, son algunas como: ¿La resistencia del fotosensor decayó? ¿El voltaje del sensor ultrasónico se elevó más allá de un umbral? ¿La salida del detector de proximidad de infrarrojo - cercano cambió de un nivel bajo a uno alto?

De cualquier forma, aun con estas limitaciones, es posible conseguir que un robot posca muchas habilidades. Un ejemplo de esto es el Robart II del Centro de Sistemas Navales Oceánicos de E.U., un robot de cinco pies de alto con sensores de sonar, infrarrojos, de detección de colisión, de movimiento por microondas, cámara de vigilancia y sensores para inundaciones y terremotos. Robart II es capaz de patrullar un edificio en busca de intrusos y avisar de distintas situaciones de alarma. También puede encontrar su estación de recarga y conectarse a sí mismo cuando así lo requiera. Otro ejemplo de la complejidad que alcanza el sistema de percepción en robots móviles autónomos es Attila, del Instituto Tecnológico de Massachusetts, un robot de seis piernas equipado con más de 60 sensores, 23 motores y 11 computadoras [1].

Mientras que el primer impulso lleva a pensar que entre mayor sea el número de sensores con que un robot es equipado, mayor será la complejidad del software que requiera para su control, esto no es necesariamente cierto. En la mayoría de los casos un gran número de sensores implica también gran independencia entre ellos, lo que significa que cada sensor requiere de relativamente poco software para producir una señal de salida a alguno de los actuadores del robot. No todos los sensores controlan la misma sección del robot, sino que se destinan a producir una reacción muy específica sólo en uno o algunos actuadores en particular.

Se verán varios tipos de sensores simples, pero antes, han de entenderse los puntos importantes que deben analizarse para implementar la interfaz de los mismos con un microprocesador.

Existen, para todo tipo de sensores, dos conceptos de importancia: la sensibilidad y el rango. Estas dos características de los sensores se determinan al momento de su fabricación, por los materiales utilizados o por la configuración física de cada sensor. Algunos sensores están diseñados para variar una o las dos de estas características.

El rango de un sensor es el conjunto de magnitudes de una cantidad física que el sensor es capaz de medir. Por ejemplo, el rango de un sensor de distancia puede ser de entre 0.15 (m) mínimo y 1.5 (m) máximo, mientras que el rango de otro sensor, también de distancia, puede estar entre los 3 (m) mínimo y 30 (m) como máximo. El primer sensor no responderá a estímulos que se encuentren más allá de los 1.5 (m), el segundo sensor sí lo hará y medirá también distancias que alcancen hasta los 30 (m). De igual forma, el segundo sensor no funcionará si trata de medir distancias menores a 3 (m), mientras que el primer sensor logrará medir distancias tan pequeñas como 15 (cm) pero no menores.

La sensibilidad es una medida del grado con el cual la señal de salida de un sensor cambia con relación al cambio de la cantidad física medida por el mismo. Si llamamos a la salida del sensor  $r$  y a la cantidad medida  $x$ , podemos definir la sensibilidad  $S$  del sensor como:

$$\frac{\Delta r}{r} = S \frac{\Delta x}{x}$$

Así, un pequeño cambio en la cantidad  $\Delta x$ , se relaciona con un pequeño cambio en la salida  $\Delta r$ , afectado por la sensibilidad  $S$ . Si la sensibilidad es mayor, el cambio en la salida provocado por un mismo cambio en la cantidad medida también será mayor. Es necesario, para el diseñador, que los sensores elegidos para un robot presenten una sensibilidad adecuada a los propósitos del diseño, ya que un sensor demasiado sensible provocará falsas lecturas mientras que un sensor poco sensible no responderá a los estímulos que se le presenten cuando así se requiera.

Debe recordarse, además, que al conectar la señal de salida de un sensor al microprocesador ésta debe ser condicionada, de modo que su nivel de entrada hacia el microprocesador se mantenga en un rango de 0 a 5 (V). Si este nivel de voltaje se convierte a un valor digital, por medio del convertidor A/D integrado al microprocesador, quedará en la forma de un valor de 8 bits, es decir, un valor discreto que se encuentre entre 0 y 255, proporcional al nivel original (este valor no posee unidades, pues es sólo la representación de una cantidad, no una cantidad física en sí misma).

Una vez que un conjunto de sensores se ha seleccionado y el circuito de interfaz adecuado se ha diseñado para conectar los sensores al microprocesador, es necesario que el microprocesador se programe para leer los sensores. Estas piezas de código se escriben en lenguaje ensamblador y se conocen como controladores por software. Los controladores por software son pedazos de código que proveen una interfaz bien definida entre un dispositivo de hardware y un programa que requiere utilizar ese dispositivo. Los controladores por software se relacionan con la interfaz de hardware-software. Estas



rutinas deben, por ejemplo, monitorear constantemente un *pin* del convertidor A/D, esperando a que un sensor se dispare; o tal vez, sea implementada una interrupción que será llamada sólo cuando cierta señal, como la de un sensor ultrasónico, alcance un nivel alto. El código o rutina del controlador de ese sensor debe recoger la lectura del sensor y guardarla en una localidad de memoria en el microprocesador. De esta forma, la salida de los sensores puede ser manejada como el valor de una variable o como una bandera dentro de un programa. Así, esa información se convierte a un nivel más alto de abstracción. Entonces otra parte del sistema de inteligencia utilizará esa bandera o variable para provocar cierto comportamiento, o tal vez primero combine esa información con alguna otra proveniente de algún tipo de 'sensor virtual', es decir, información derivada de cierto proceso dentro del programa.

En conclusión, es necesario cuidar tanto las características de los sensores a utilizar, como la forma en que la salida de un sensor se prepara para ser introducida al microprocesador (por hardware). En ocasiones será necesario usar 'amplificadores' de voltaje, que aumenten o disminuyan el nivel de la señal; otras veces se requerirán escalas de conversión no lineales, pues la magnitud de la variable física que se desea medir puede variar, tal vez, con una relación exponencial en vez de lineal, etc.

#### 4.1.2 Tipos De Sensores

A continuación se mencionan brevemente los distintos tipos de sensores que pueden ser útiles en el diseño y la implementación de robots móviles, su funcionamiento, ventajas y limitaciones.

**Fotosensores (sensores de luz).**- Un sensor de luz habilita comportamientos de un robot tales como seguir un camino delimitado en el suelo de una fábrica, dirigirse hacia una señal luminosa o reconocer ciertos patrones (como códigos de barras) que le den información. Sensores de luz sencillos son llamados fotoresistencias, que son resistencias que varían su valor de acuerdo a la luz que reciben. Un cambio en la resistencia provoca cambios de voltaje que pueden ser interpretados por el microprocesador. Otra opción son los fototransistores o los fotodiodos, cuyo principio de funcionamiento es el mismo. Este tipo de sensores generalmente requieren una interfaz sencilla y responden rápidamente, con alta sensibilidad y buen rango. Su limitante es que el medio ambiente en que se utilizan debe estar bien controlado para no provocar falsas lecturas. Una ventana por la que entren los rayos del sol puede alterar el contraste entre el suelo de una fábrica y el camino que el robot debe seguir, imposibilitando que el sensor de luz distinga a uno del otro.

**Sensores de proximidad por infrarrojo.**- Una manera para lograr que un robot siga a una persona o recorra un camino guiado por un muro, a una distancia determinada, es equipándolo con un sensor de proximidad de infrarrojo cercano. Estos sensores no reportan la distancia a la que los objetos en su rango se encuentran, sólo distinguen cuando algo está presente o no frente a ellos. Se les conoce como IRs, por sus siglas en inglés, aunque en un sentido estricto no son capaces de detectar las longitudes de onda mayores del infrarrojo (el calor). En vez de ello, son sensibles en el rango del infrarrojo que se encuentra justo debajo de la luz visible. Pueden encontrarse emisores y detectores de infrarrojo por separado, o sensores con ambos componentes y ciertos adecuadores de la salida integrados en un solo dispositivo. Los sensores de proximidad por infrarrojo cercano requieren una interfaz algo compleja, pues es necesario asegurar que la frecuencia del emisor y del detector se mantengan estables y evitar falsas lecturas. Sin embargo, existen controladores implementados en circuitos integrados que resuelven este problema. Debe cuidarse que el sensor infrarrojo no sea expuesto a luces fluorescentes, pues ello provocaría mucho ruido en los detectores. Los sensores infrarrojos poseen un haz de detección más angosto que el de otros sensores como los medidores de distancia por sonar, lo cual es conveniente. Sus inconvenientes son que no funcionan bien si las superficies que buscan detectar son muy irregulares o se encuentran en determinados ángulos con relación al haz de detección. Los sensores infrarrojos tampoco son convenientes cuando el movimiento del robot se produce en más de un plano, pues ocurren falsas lecturas con facilidad.

**Sensores de temperatura.-** Uno de los tipos de sensores más útiles para lograr la interacción de los seres humanos con robots son los sensores de temperatura. Es uno de los componentes esenciales en algunos tipos de alarma de detección de 'movimiento'. El elemento activo en este tipo de sensores es usualmente un cristal de tantalio o litio. Una carga es inducida cuando el cristal se calienta, de modo que puede detectarse la radiación infrarroja en el rango de 8 a 10 ( $\mu\text{m}$ ) (de la energía emitida por el cuerpo humano). El punto débil de estos sensores es que no resultan útiles para la detección de obstáculos o estructuras que se encuentren a la misma temperatura que el medio ambiente, y que si es necesario que el sensor vaya protegido, por ejemplo, con una placa de acrílico, esto impedirá que las lecturas se realicen correctamente. Por otro lado, la interfaz necesaria es mínima, y con una buena cantidad de software pueden lograrse respuestas muy interesantes de los robots.

**Sensores de fuerza.-** En general, los sensores de fuerza han probado ser los más eficientes, con el menor nivel de ruido, y sus señales son las más sencillas de interpretar. Estos sensores pueden utilizarse para determinar cuando un robot está en contacto con otro objeto y también en qué posición se encuentra ese objeto relativamente al robot. Tal información es útil cuando se desea que el robot se mantenga libre de colisiones. La variedad de sensores de fuerza abarca microswitches, resistencias detectoras de fuerza, sensores de conductividad eléctrica, potenciómetros lineales, etc. Cabe mencionar que estos sensores se encuentran limitados a aplicaciones en las que el robot no tenga una velocidad muy grande, de modo que sea capaz de detenerse en una distancia muy corta, cuando el sensor de fuerza produzca una lectura. Asimismo, los objetos que el sensor pueda detectar deben presentar ciertas características, como que posean dimensiones comparables a las del robot, que no vayan a producir daños a la estructura del mismo, etc. La interfaz necesaria consiste generalmente de uno o varios divisores de voltaje.

**Sensores de sonido.-** Los sensores de sonido pueden dividirse en dos grandes grupos: sensores que detectan sonido audible, y sensores que trabajan con ultrasonido. Los primeros se usan para que el robot interactúe con su operador, los segundos sirven para la detección y evasión de obstáculos.

Micrófonos y películas piezoeléctricas son ejemplos de sensores que detectan sonido audible. Usualmente es necesario que su salida sea amplificada y algunas veces filtrada; muchas veces también se requiere de un microprocesador en paralelo al microprocesador principal para que se encargue únicamente del monitoreo de las señales de un micrófono, pues para detección de sonidos de alta frecuencia o sonidos de corta duración es indispensable que se tomen muestras de la señal del micrófono al menos cada milisegundo y esto ocupa mucho tiempo de procesamiento. Las películas piezoeléctricas, además de sonido, pueden usarse para detección de temperatura, fuerza aplicada y vibraciones. Su principio es como el de un micrófono. Antes de utilizar cualquier sensor de sonido audible deben considerarse las fuentes de ruido, como los motores del propio robot, o los del medio ambiente. El sensor debe ser resguardado de alguna forma de estas fuentes de ruido y al mismo tiempo ser capaz de captar aquellos sonidos que se desean detectar.

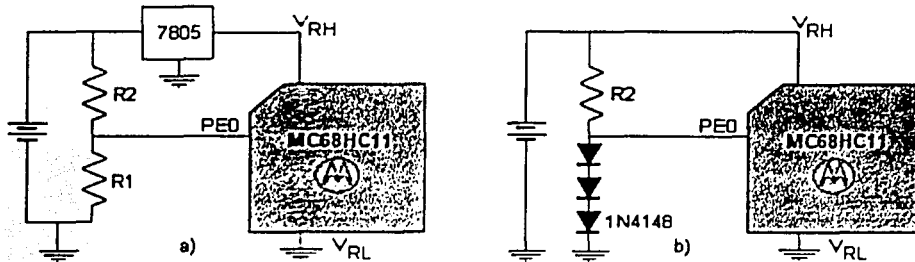
Los sensores de ultrasonido, también llamados sonares, no sólo son capaces de detectar la presencia de objetos dentro de su haz de detección, sino que también pueden medir la distancia a la que los obstáculos se encuentran. Esta característica puede utilizarse para que un robot viaje a través de corredores, para la construcción de 'mapas visuales' del robot, y para la evasión de obstáculos a una distancia considerable. En la sección 4.1.3 de este capítulo se trata más detalladamente el funcionamiento de este tipo de sensores, pues dadas sus características son los que más se adecuan a los requerimientos del proyecto.

**Sensores de posición y orientación.-** Cuando un robot se desplaza, es muy ventajoso que pueda conocer su posición o realizar ciertas mediciones con relación a su inclinación, la dirección de la gravedad o cuál es la distancia recorrida desde un punto conocido. Sensores que pueden dar ese tipo de información son brújulas, medidores de rotación de eje, giroscopios, sensores de inclinación, etc. Cada sensor tiene sus

limitaciones, por ejemplo, una brújula magnética no funcionará en ambientes donde haya campos electromagnéticos intensos, como los de motores, cableado o estructura metálica de algún edificio. Si el robot móvil no usa ruedas para su desplazamiento, un medidor de rotación de eje servirá de poco.

En el desarrollo de este proyecto se utilizó un receptor GPS, apropiado para exteriores y con una sensibilidad para establecer la posición del móvil no menor a 8 (m) en condiciones óptimas. El capítulo 2 de este trabajo fue dedicado especialmente a la descripción del sistema GPS, el receptor utilizado y las características de su señal. Un receptor GPS puede ser considerado como un sensor para establecer la posición del móvil, pero no en sentido estricto, pues a diferencia de los sensores descritos en este capítulo el GPS involucra un procesamiento de señales electromagnéticas a cargo de todo un sistema, es decir, no se trata de algo tan sencillo como realizar la medición directa de una cantidad física en la forma en que lo hace un transductor.

**Sensores internos.-** Para asegurar el correcto funcionamiento de un robot es una buena idea monitorear su estado interno, por ejemplo, la temperatura de sus componentes o el nivel de sus baterías. En la mayoría de los casos es más sencillo diseñar un sensor para las variables internas del sistema que conseguir uno que se adecue a las necesidades específicas en cada caso. Como ejemplo, la figura 4.1 muestra dos maneras para implementar la medición del nivel de una batería que requieren elementos sencillos, baratos y con una configuración muy simple.



a) Si el microprocesador trabaja con una fuente regulada, basta un divisor de voltaje para implementar un monitor de batería. b) Cuando no se tiene un regulador de voltaje, una caída de voltaje provocada por los diodos dará la referencia de voltaje necesaria. En PE0 se mide el voltaje de la batería, si este valor se encuentra por debajo de un nivel determinado, una señal puede ser accionada.

Figura 4.1 Dos Maneras de Implementar un Monitor de Batería

#### 4.1.3 Funcionamiento De Los Sensores Ultrasónicos

De entre los tipos de sensores disponibles, los que mejor cumplen con los requerimientos de este proyecto son los sensores ultrasónicos. Los sensores ultrasónicos miden distancias utilizando un transductor que envía señales de ultrasonido. Cada señal producida por el transductor contiene una serie de pulsos de ultrasonido, esta señal se dispersa en forma de cono, se refleja en una superficie y regresa al transductor ocasionando una detección. El tiempo que requiere la señal ultrasónica para viajar hasta la superficie de un objeto y regresar al transductor es medido y convertido en una distancia, dado que se conoce la velocidad a la que viaja la señal ultrasónica.

La detección de objetos y la medición de distancias por medio de ultrasonido son afectadas por varios factores, como el tamaño de los objetos detectados, el tipo de superficie, el ángulo en el que se encuentran los objetos respecto a la emisión de la señal ultrasónica, y la distancia a la que se encuentran del transductor.

Si un objeto plano y liso está inclinado más de  $3^\circ$  respecto a la normal del eje del haz, parte de la señal se reflejará fuera del área de detección del sensor y el rango de detección disminuirá. Para objetos pequeños localizados cerca del sensor, la desviación respecto a la normal puede incrementarse hasta  $8^\circ$ . Si un objeto se inclina más de  $10$  o  $12^\circ$  respecto a la normal, toda la señal se reflejará fuera del alcance de detección del sensor y éste no responderá correctamente.

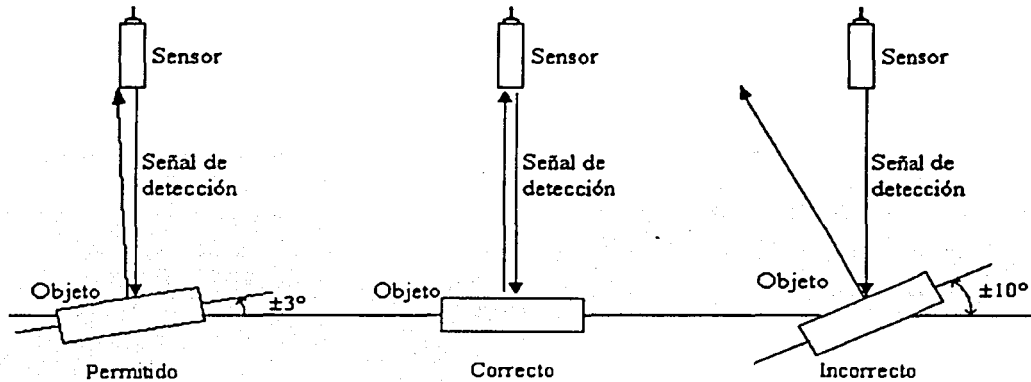


Figura 4.2 Máximas Inclinaciones Para Objetos Planos y Lisos

Los sensores ultrasónicos emiten un haz de sonido que se encuentra dentro de un ángulo de cono de haz donde se eliminan los lóbulos laterales. El tamaño del objeto detectado es importante con relación al punto de emisión del haz. Teóricamente, el objeto más pequeño que puede detectar un sensor ultrasónico tiene un tamaño de la mitad de longitud de onda de la señal transmitida. Por lo general, los objetos son más grandes y son detectados a varias distancias. Para poder estimar el área de cobertura de la señal ultrasónica a cierta distancia, se utiliza la fórmula:

$$B_{cx} = 2X \tan(\delta/2)$$

Donde:

- $B_{cx}$ .- Es el diámetro del cono de haz a la distancia  $X$
- $X$ .- Es la distancia que existe entre el objeto y el sensor
- $\delta$ .- Es el Ángulo del Cono de Haz

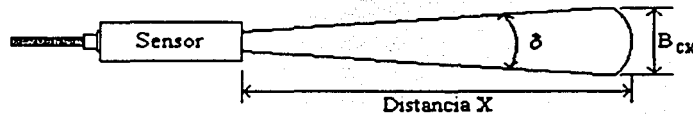
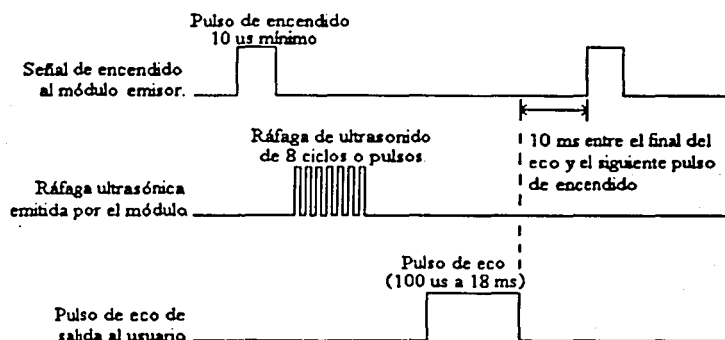


Figura 4.3 Estimación del Área de Cobertura Para un Sensor Ultrasónico

Los valores de los ángulos del cono de haz se ubican donde el haz es atenuado en 3 dB, fuera de este ángulo de cono existe señal ultrasónica que es muy débil y es probable que los objetos no sean detectados. El patrón del haz producido por un sensor ultrasónico se expresa mediante el número de grados que el haz se dispersará desde la línea central del sensor. Aunque el haz ultrasónico continúa dispersándose hasta desaparecer, el área de detección del sensor comienza a cerrarse de acuerdo al rango del sensor. El área de detección es afectada por el número de pulsos transmitidos en una ráfaga de la señal ultrasónica y por el nivel de sensibilidad del sensor.

Los sensores ultrasónicos operan bajo un sistema de medición sujeto a un tiempo determinado (ventana de tiempo). Cuando se ajusta la sensibilidad del sensor se establece también una ventana de tiempo para la medición. El sensor tomará en cuenta los ecos recibidos durante dicha ventana de tiempo; como el eco del fondo detrás del objeto tarda más en llegar al sensor, no es tomado en cuenta dentro de la medición.



Si no se detecta un objeto, cada pulso de eco es de approx 36 ms

Figura 4.4 Temporización de las Señales de un Sensor Ultrasónico

La frecuencia máxima a la cual el sensor es capaz de cambiar de modalidad es llamada frecuencia máxima de conmutación y depende de muchas variables. Las variables más significativas son el tamaño del objeto, el material del objeto y la distancia hacia el objeto. Mientras el objeto sea más pequeño, más difícil será su detección. La frecuencia máxima en el caso de tener un objeto pequeño será menor que en el caso de tener un objeto grande. Los materiales que absorben sonidos de alta frecuencia como el algodón, esponjas, etc., son más difíciles de detectar que el acero, vidrio o plástico. Así pues, ellos también tienen una frecuencia máxima menor.

La distancia entre el objeto y el sensor también es muy importante para determinar la frecuencia máxima de conmutación. El sensor emite un haz ultrasónico a través del aire, que toma tiempo en dejar el sensor, viajar hasta el objeto, golpear al objeto y regresar al sensor como eco. Mientras más retirado se encuentre el objeto, mayor es el tiempo que le toma a la señal en realizar este ciclo y menor es la frecuencia de conmutación.

Se llama zona muerta a la zona en donde el sensor no puede detectar los objetos con precisión. Es la distancia entre la cara del sensor y la distancia mínima de detección. Si el objeto se encuentra muy cerca del sensor, el pulso emitido para detectar los obstáculos llegará al objeto y regresará su eco antes de que todo el pulso sea transmitido por completo. El sensor ignora las señales de eco que le lleguen mientras no esté en la modalidad de receptor.



Figura 4.5 Zona Muerta en un Sensor Ultrasónico

Entre las condiciones ambientales que afectan las mediciones se encuentran la temperatura, humedad, presencia de gases y la presión. La mayoría de los sensores ultrasónicos se encuentran equipados con mecanismos de ajuste manuales o automáticos para compensar varias de esas condiciones.

La velocidad del sonido depende de la temperatura, por lo que un sensor interno de temperatura adapta la frecuencia de reloj del contador de la ventana para ayudar a compensar las variaciones de la temperatura del aire. Además, las fluctuaciones de temperatura en el recorrido del haz pueden causar dispersión y refracción de la señal ultrasónica, afectando la precisión y estabilidad de la medición. Si es necesario detectar objetos con altas temperaturas, se recomienda ubicar al sensor y al objeto sobre un plano horizontal de manera que el sensor esté direccionado hacia la parte del objeto con una temperatura menor.

Una señal ultrasónica presenta una vibración a una frecuencia mayor que el rango de audición humana, usualmente esto es mayor a los 20 KHz. La mayoría de los sensores ultrasónicos utilizan un único transductor para transmitir y recibir la señal de ultrasonido, y operan típicamente a frecuencias entre los 40 KHz y los 250 KHz.

Los sensores ultrasónicos se usan comúnmente para una gran variedad de mediciones de presencia, de proximidad o de distancia, sin contacto. La variedad de sensores en el mercado difieren unos de otros en su configuración de montaje, su resistencia a las condiciones medioambientales, y sus funciones electrónicas. Acústicamente, operan a diferentes frecuencias, y presentan también diferentes patrones de radiación. Usualmente no es difícil seleccionar un sensor que se adecue a los requerimientos de una aplicación particular.

El sistema de sensores seleccionado para el desarrollo del proyecto es un equipo originalmente diseñado para usarse mientras se estaciona un auto. Los sensores ultrasónicos detectan un objeto cercano a la parte trasera del auto (una pared, otro auto, una persona, etc.) y emiten una señal que alerta al conductor del mismo.

#### 4.1.4 Características De Los Sensores Ultrasónicos Empleados

El equipo de sensores ultrasónicos utilizados para la implementación del prototipo del sistema de navegación es el modelo K3502 PARKING RADAR distribuido por la compañía Jameco. Es un equipo económico y con un rango de medición ajustable entre 5 (cm) y 1.5 (m). Este rango es adecuado para el diseño e implementación de prototipos de un sistema de navegación pero no para la implementación de un sistema real, pues resulta muy pequeño comparado con el rango que se requiere para el tipo específico de aplicación (el rango necesario para un sistema de navegación como el que se propone debe encontrarse alrededor de los 15 (m) como mínimo).

Los sensores K3502 poseen dos transductores de ultrasonido: un transmisor que emite los pulsos ultrasónicos y un receptor que capta el eco de rebote y transforma este eco en una señal eléctrica. La figura 4.6 ilustra la forma y dimensiones de uno de estos transductores; tanto el transmisor como el receptor comparten estas características.

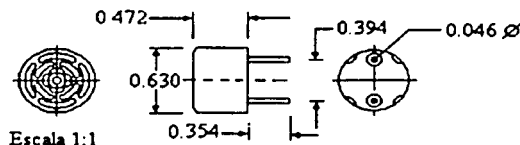


Figura 4.6 Diagrama Técnico de un Transductor Ultrasónico

Cada par de transductores, más una tarjeta de control, componen un sensor ultrasónico completo. Estos sensores proporcionan una señal de salida de 10(V) mientras no se detecte la presencia de algún obstáculo. En el momento en que se presenta una detección positiva el equipo entrega como señal de salida un tren de pulsos de 10 (V) a 26 (Hz) con una duración de cada pulso de 150 ( $\mu$ S), esta señal entra a un buzzer que genera un sonido audible para indicar al usuario que un objeto se ha detectado. La figura 4.7 presenta el sensor armado en dos tarjetas: la tarjeta de los transductores y la de control, donde puede apreciarse el buzzer de salida acústica.

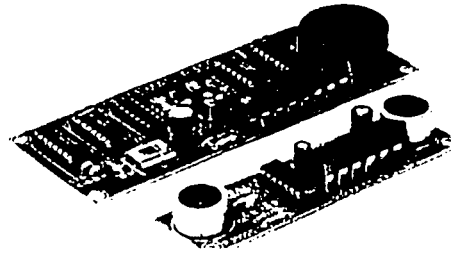


Figura 4.7 Sensor Ultrasónico

La figura 4.8 muestra el diagrama eléctrico de los sensores ultrasónicos en su conjunto. El dispositivo denominado SENS1 es el transmisor, SENS2 el receptor y la salida del circuito se encuentra en la terminal negativa del elemento denominado BUZ.

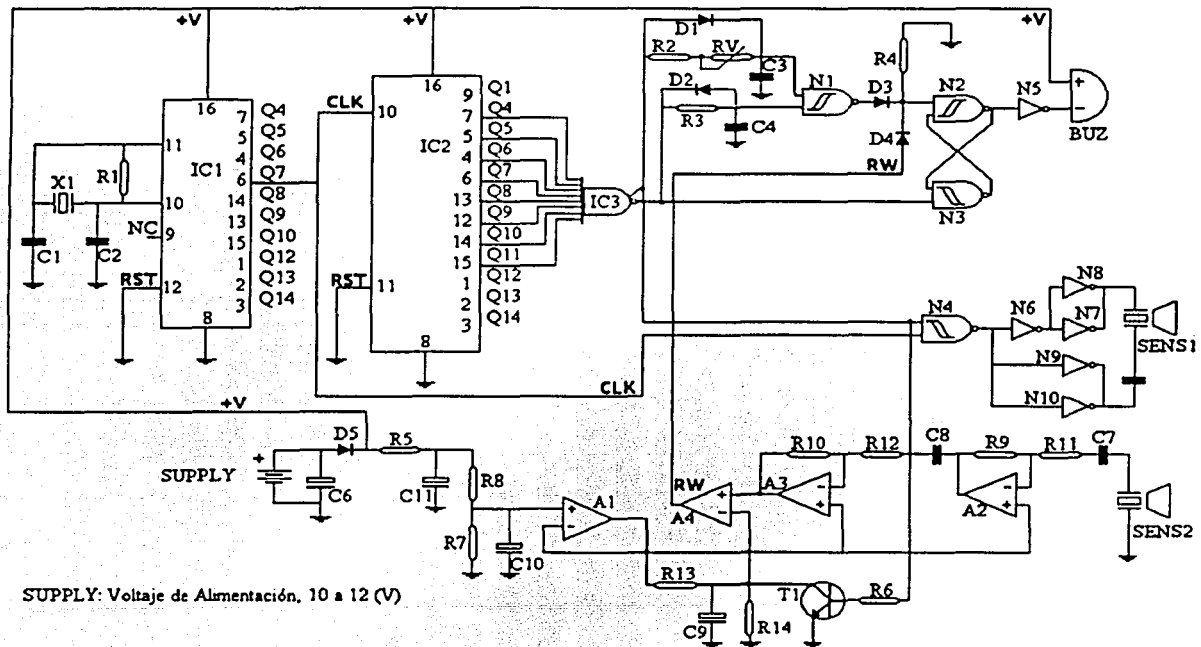


Figura 4.8 Diagrama Eléctrico de los Sensores Ultrasónicos

El circuito de interfaz necesario para conectar la salida de estos sensores al microprocesador consiste en un inversor de colector abierto (IC 7405) para tener una señal positiva que pueda leerse en el puerto del microprocesador, un regulador de voltaje a 5 (V) (IC 7805) para mantener la señal en un nivel TTL para el puerto y un interruptor analógico (IC 4016) que sirve para deshabilitar la señal de los sensores en ciertos momentos de la ejecución del programa principal del sistema (refiérase al capítulo 6, sección 6.3.3).

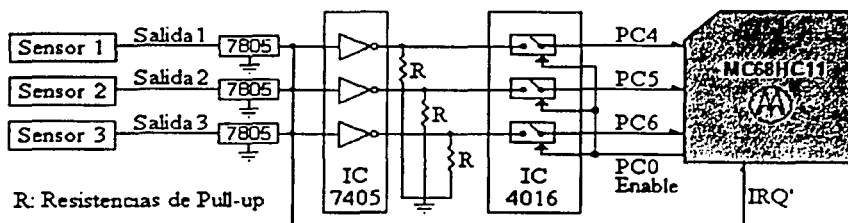


Figura 4.9 Circuito de la Interfaz Sensores - Microprocesador

Para el sistema de navegación se eligió una configuración con tres sensores ultrasónicos que proporcionarara un área de cobertura de 180° al frente del móvil, tal como muestra la figura 4.10.

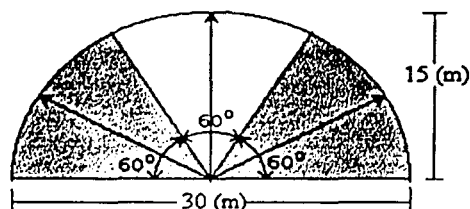


Figura 4.10 Área de Cobertura del Sistema de Sensores Ultrasónicos

Se propone utilizar sensores ultrasónicos con un rango de detección de aproximadamente 15 (m) y un ángulo de apertura de haz de 60° cada uno. Equipos con estas características se encuentran disponibles en el mercado. Sin embargo, no es práctico trabajar con este tipo de equipo durante la fase de diseño, pues su costo es más elevado que el de los sensores ultrasónicos de menor rango de detección y se requiere mayor espacio para realizar pruebas. La figura 4.11 presenta el patrón de haz de detección del sensor ultrasónico utilizado. La tabla 4.1 condensa las especificaciones técnicas del mismo.

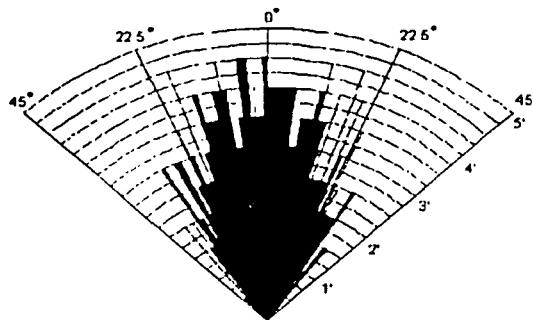


Figura 4.11 Patrón del Haz de los Sensores Ultrasónicos



| <b>Características Técnicas del Transmisor</b>   |                              |
|--|------------------------------|
| Tipo   | Murata/Eric P/N 492948.      |
| Frecuencia Nominal                               | 40 (KHz.)                    |
| Máxima sensibilidad (nivel de sonido)            | 110 dB.                      |
| Ancho de banda                                   | 4 (KHz) mínimo a 100 dB.     |
| Capacitancia                                     | 1300 (pF)                    |
| Resistencia                                      | 100 (M $\Omega$ ) mínimo.    |
| Variación de sensibilidad respecto a temperatura | -10 dB desde -20°C a 80° C   |
| <b>Características Técnicas del Receptor</b>     |                              |
| Tipo   | Murata/Eric P/N 492951.      |
| Frecuencia nominal                               | 40 (KHz).                    |
| Máxima sensibilidad (nivel de sonido)            | -67 dB mínimo.               |
| Ancho de banda                                   | 6 (KHz) a -74 dB.            |
| Capacitancia                                     | 2000 (pF.)                   |
| Resistencia                                      | 1 (M $\Omega$ ) mínimo.      |
| Variación de sensibilidad respecto a temperatura | -10 dB desde -20°C a 60°C.   |
| <b>Para el Sistema de Sensores en Conjunto</b>   |                              |
| Rango de medición                                | 5 (cm) – 1.5 (m) (ajustable) |
| Frecuencia de transmisión                        | 40 (KHz)                     |
| Frecuencia de muestreo                           | 26 (Hz)                      |
| Fuente de alimentación                           | 10- 15 (Vdc)/ max. 16 (mA)   |

**Tabla 4.1 Especificaciones Técnicas de los Sensores Ultrasónicos**

## **4.2 SISTEMAS DE PROPULSIÓN Y DIRECCIÓN**

Existe una gran variedad de maneras para moverse a través de una superficie sólida, desde arrastrarse hasta saltar. Para los robots terrestres las tres maneras más comunes son a través de ruedas, piernas y orugas. Si el móvil o vehículo es aéreo, se cuenta con hélices, turbinas, alas y globos. Cuando el móvil debe desplazarse en el agua debe pensarse en dos opciones diferentes: sobre y dentro del agua. También allí hay una gran variedad de mecanismos de desplazamiento, desde el uso de aletas o propelas hasta turbinas submarinas.

Buscando que un móvil se desplace sobre el agua, puede observarse que la manera más sencilla y barata es por medio de una propela y un timón. Sólo se requiere de un motor de propulsión y otro motor que ayude a controlar la dirección del móvil. Por supuesto, la precisión del movimiento en el agua es mucho menor que la de un movimiento en tierra. El sólo tratar de mantenerse en una posición fija sobre o dentro del agua significa una tarea extraordinaria para un robot acuático, incluso sin considerar corrientes en el medio. Pese a ello, existen muchas aplicaciones al alcance de robots o móviles que se desplacen por el agua. Este proyecto trata sólo sobre una de ellas en particular. A continuación, se describen los sistemas de propulsión y dirección utilizados en el desarrollo del proyecto. Cabe aclarar que se trata, como en el caso del sistema de sensores, tan sólo de modelos para experimentación, es decir, que no presentan las características específicas requeridas para ser implementados en un vehículo real (dimensiones, potencia, engranaje, etc.).

#### 4.2.1 Control Del Motor De Propulsión Del Vehículo

Un motor eléctrico convierte energía eléctrica en energía mecánica. Los motores se encuentran en gran variedad de formas y tamaños. Existen motores electromagnéticos de corriente directa (DC) y motores electromagnéticos de corriente alterna (AC) con un número de variantes de cada tipo. Los motores de AC son usados generalmente para maquinaria grande (herramientas mecánicas, lavadoras, secadoras, etc.) y se alimentan de la toma de corriente alterna. Podemos encontrar motores de AC de fase sencilla, trifásicos, de polo acanalado, etc. cada uno de ellos tiene aplicaciones muy específicas. Los motores de AC son poco usados en robots móviles pues la fuente de poder de los robots es usualmente una batería de DC [2].

Los motores de DC son utilizados para trabajo ligero, y se adecuan bien a la implementación de modelos y prototipos. Aparecen también en una gran variedad de formas y tamaños: motores con núcleo permanente de hierro, imán permanente con rotor sin hierro, de imán permanente sin cerdas, de pasos con imán permanente, motores de pasos híbridos, etc.

Para las necesidades de un robot, un motor de DC posee demasiada velocidad y poco torque, así que el motor debe ser engranado de manera que su torque aumente y su velocidad disminuya. Los motores adecuados para aplicaciones en robots son de imán permanente y rotor sin hierro, debido a que son muy ligeros. La mayoría de estos motores cuentan con dos terminales. Al aplicar un voltaje a estas terminales se provoca la rotación del motor en una dirección; cuando se aplica el mismo voltaje con la polaridad inversa, el motor girará en la dirección contraria. La polaridad del voltaje determina la dirección de giro, mientras que la magnitud o amplitud del voltaje determina la velocidad del motor. La corriente que circula por el motor, en una u otra dirección, es proporcional al torque o esfuerzo que el motor desarrolla.

Un microprocesador no puede manejar un motor directamente, dado que no proporciona suficiente corriente. En vez de ello, debe existir una interfaz de hardware de modo que la alimentación del motor provenga de una fuente externa mientras que el microprocesador se encarga sólo de las señales para el control. Este circuito de interfaz puede ser implementado de muchas maneras, utilizando relevadores, transistores bipolares, transistores de efecto de campo (MOSFETs por sus siglas en inglés), y circuitos integrados. Independientemente de la tecnología utilizada, la topología del circuito es básicamente la misma.

Este tipo de circuitos se conoce como de 'puente H' y consiste de un conjunto de cuatro interruptores conectados formando una H, donde las terminales del motor que se busca controlar ocupan la barra central de la H, tal como muestra la figura 4.12.

Cada interruptor es implementado con un relevador o un transistor, donde la energía es suministrada por una batería y la señal de control por el microprocesador. Como ejemplo, si los interruptores S1 y S4 en la figura 4.12 son cerrados mientras que S2 y S3 permanecen abiertos, la corriente fluye de izquierda a derecha a través del motor. Cuando se cierran S2 y S3 mientras S1 y S4 se abren, la corriente fluye de derecha a izquierda, haciendo que el motor gire en reversa. Si ninguno de los switches se cierra, el motor no estará polarizado, y si todos los interruptores se cierran, el motor no se moverá o, en su caso, frenará.

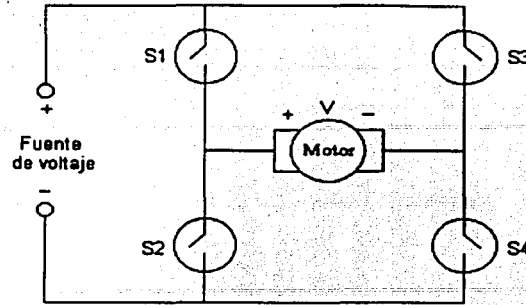
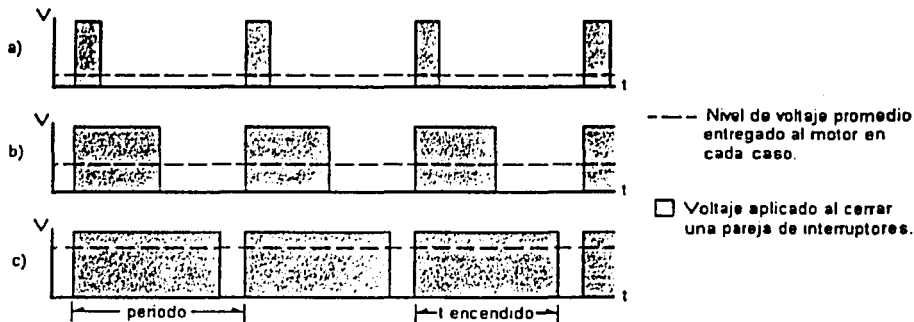


Figura 4.12 Topología del Circuito de Puente H

Para controlar la velocidad del motor los interruptores del puente H se cierran y abren coordinadamente a diferentes frecuencias con el fin de aplicar un voltaje promedio variable a las terminales del motor. La figura 4.13 ilustra esta técnica llamada 'modulación por ancho de pulso' o, por sus siglas en inglés PWM.  $V$  es el voltaje en las terminales del motor,  $t$  es el tiempo sobre el que se mide la duración de las señales hacia los interruptores.

La velocidad del motor puede ajustarse al cambiar el ciclo de trabajo del tren de pulsos:

$$\text{Ciclo de trabajo} = \frac{t_{\text{encendido}}}{\text{Periodo}} \times 100$$



Encender y apagar los interruptores en el puente H con diferente duración, crea un voltaje promedio en las terminales del motor. Entre mayor sea el tiempo de encendido, mayor es el voltaje promedio entregado.

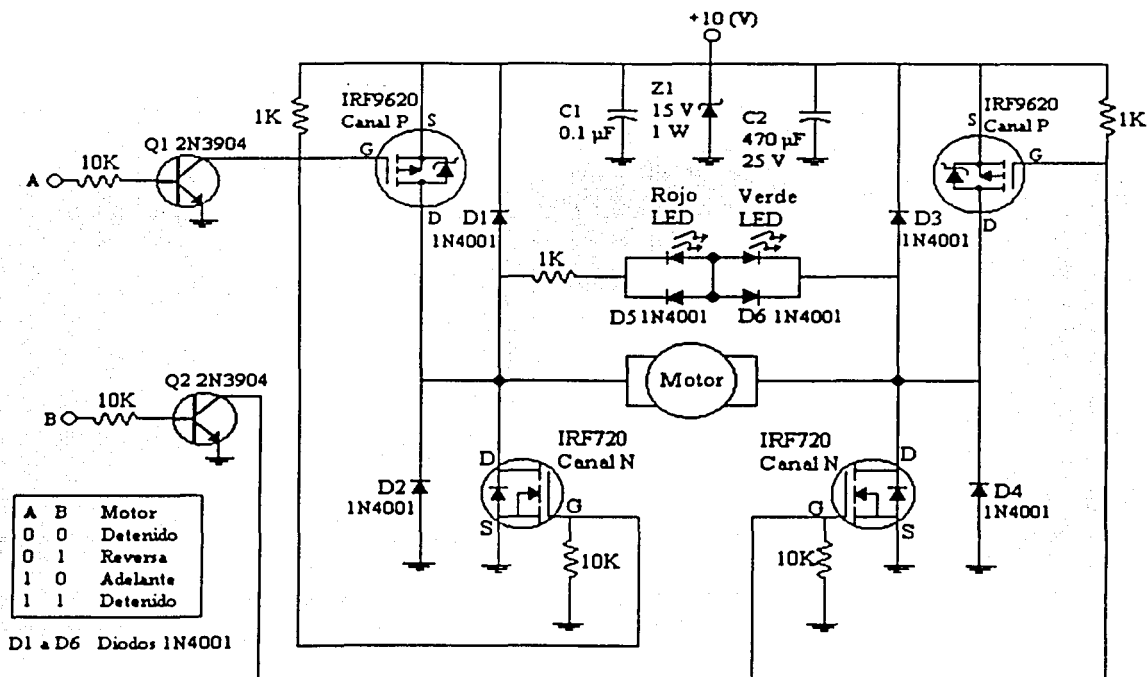
a) Ciclo de trabajo de 15% con respecto al periodo    b) Ciclo de trabajo de 50%    c) Ciclo de trabajo de 85%

Figura 4.13 Modulación de Voltaje por Ancho de Pulso

Los interruptores ideales de la topología de puente H deben sustituirse por dispositivos reales. Los relevadores usualmente no cambian de estado tan velozmente como se requiere, así que la solución está en los transistores bipolares de juntura o en los MOSFETs. Otra solución es conseguir un circuito integrado de puente H, disponible en el mercado.

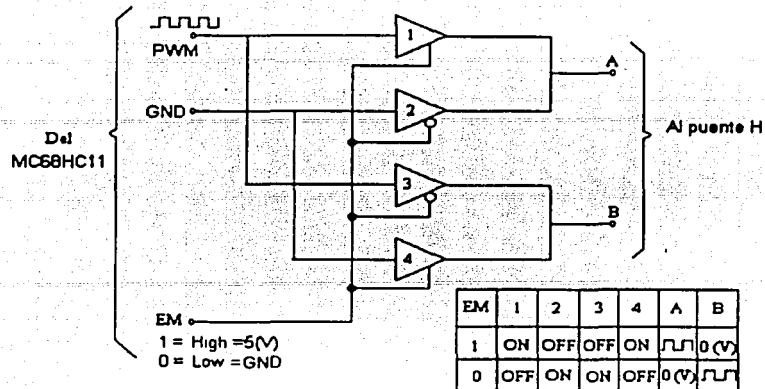
La figura 4.14 muestra el diagrama del puente H implementado para este proyecto. Este circuito requiere de dos entradas, A y B para controlar el motor. El microprocesador MC68HC11 dispone de varias salidas de tren de pulsos con ciclo de trabajo variable, sin embargo, resulta más fácil conmutar una sola señal por medio de hardware que programar la habilitación coordinada de dos salidas de PWM hacia el puente H.

La figura 4.15 muestra el diagrama del circuito que hace esto posible. En caso de que no se quiera cambiar la dirección del motor, el circuito accesorio no es necesario; puede conectarse la señal de PWM del microprocesador directamente a la base de alguno de los transistores TBJ, A o B. La base del otro transistor debe llevarse a tierra. Si se tienen problemas de espacio para la implementación del puente H y el circuito de conmutación, se preferirá el uso de controladores en circuitos integrados. El esquema en la figura 4.16 presenta esta solución.



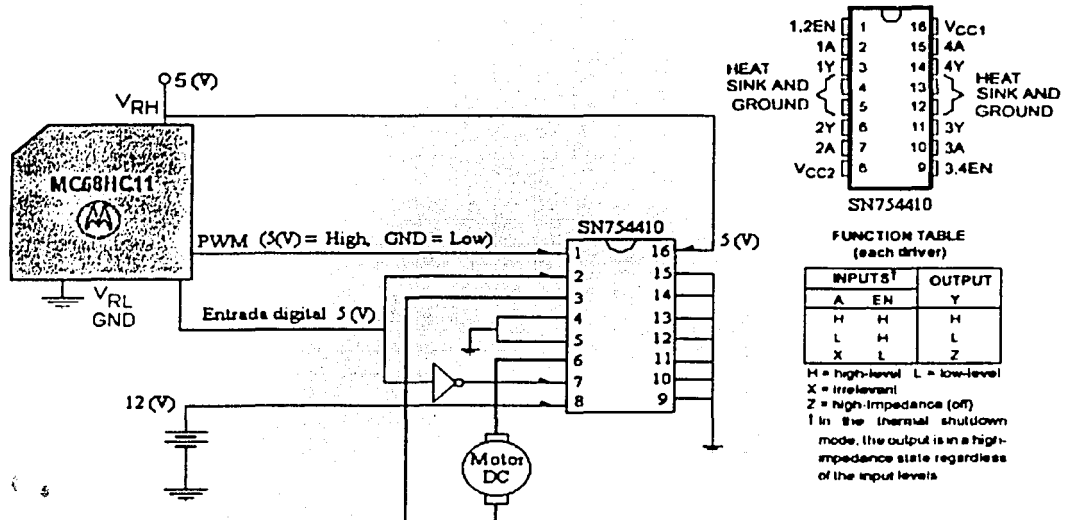
En este circuito, los interruptores ideales del puente H fueron reemplazados por MOSFETs, se agregaron dos capacitores y un diodo zener para protección del circuito, y el arreglo de LEDs y diodos en paralelo con el motor es para señalización. La señal de PWM se aplica en la base de los transistores A y B, de acuerdo a la tabla mostrada.

Figura 4.14 Diagrama Eléctrico del Circuito de Control del Motor de Propulsión



El circuito recibe las señales del microprocesador, y por medio de buffers de tres estados (LS74125A buffer inversor y LS74126A buffer no inversor) es capaz de conmutar estas señales de acuerdo con la tabla que se muestra. EM es la señal de habilitación para los buffers, en la tabla también se muestran los estados de los buffers en cada caso

Figura 4.15 Circuito de Conmutación para las Entradas de Control del Motor de Propulsión



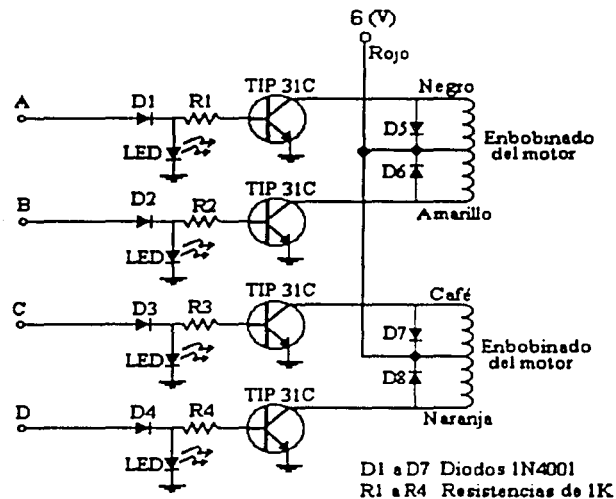
Las entradas a los pines 2 y 7 del integrado en el diagrama definen la dirección de giro del motor de acuerdo con la tabla mostrada y el esquema de pines del circuito. El microprocesador entrega la señal de PWM al integrado mientras que la alimentación para el motor de DC es externa, de una batería de 12 (V).

Figura 4.16 Control de un Motor de DC por PWM Usando el CI SN754410

#### 4.2.2 Control Del Motor De Dirección Del Vehículo

Otro tipo de motores de DC, los motores de pasos o motores a pasos, tienen más de dos terminales eléctricas, tantas como seis u ocho. Las señales, o comandos que se aplican a estas terminales energizan las diferentes bobinas del motor. Para que el motor gire continuamente las bobinas deben ser energizadas secuencialmente en un orden determinado. El tiempo entre las señales que se ingresan al motor determina la velocidad del mismo, y el orden de la secuencia determina la dirección de giro. Una característica especial de estos motores es que su posición puede ser controlada con gran precisión dependiendo del número de señales que se le den. Esta característica hace que se utilicen en el control de mecanismos que requieran mantener una posición determinada o cambiar a una o varias de ellas.

Un dato que debe conocerse para el control del motor de pasos es la magnitud de sus 'pasos', es decir, cuántos grados gira por cada comando que se le da. En este proyecto se utilizó un motor de pasos unipolar de 7.5 grados por paso, para controlar el timón del vehículo marítimo. La interfaz del motor de pasos es muy sencilla, consiste de cuatro transistores TIP31C que trabajan como interruptores, de manera que una batería externa se encargue de alimentar al motor de pasos de cuatro bobinas, proporcionándole la corriente necesaria que el microprocesador por si mismo no es capaz de dar. La figura 4.17 muestra un



Las entradas de control A,B,C,y D provienen del puerto B del microprocesador, el motor de pasos utilizado en este caso tiene cuatro bobinas, cada una de ellas en paralelo con los diodos D5,D6,D7 y D8.

diagrama de la interfaz de hardware, o controlador del motor de pasos.

Figura 4.17 Circuito de Control Para el Motor de Dirección

Las entradas de control del circuito en la figura 4.17 vienen del puerto B del microprocesador, estas entradas cambian secuencialmente en un orden determinado para hacer girar el motor de pasos en cierto ángulo de acuerdo a las necesidades instantáneas del móvil, logrando así que el mismo cambie de rumbo cuando lo requiera. Por ejemplo, la siguiente secuencia en los *pins* del puerto B: 0001 0010 0100 1000 (1 2 4 8 en sistema decimal), harán que el motor se mueva 21 grados en el sentido de las manecillas del reloj. Si se alimenta la secuencia invertida: 1000 0100 0010 0001 (8 4 2 1), el motor girará en contrasentido. Los LEDs presentes en el circuito de control sirven de señalización, y los diodos en las entradas a los transistores protegen el puerto B del microprocesador, que es únicamente de salida.

**REFERENCIAS BIBLIOGRÁFICAS**

- [1] Proyecto Micro-Rober (robots Hannibal y Attila); Massachussets Institute of Technology;  
<http://www.ai.mit.edu/projects/humanoid-robotics-group/retired-robots/retired-robots.html>
- [2] Jones, Joseph L. & Flynn, Anita M.; "Mobile Robots Inspiration to Implementation"; pp. 161 – 225;  
1993; AK Peters Ltd.

## ***Algoritmos de Navegación Autónoma***

Se entiende el término 'Navegación' como el desplazamiento controlado de un móvil a través de un espacio definido en el cual es posible la presencia de obstáculos. La navegación autónoma existe cuando el sistema de navegación del móvil no recibe señales de control externas sino que produce sus propias señales de control internamente. Dependiendo de las características del espacio por el cual el móvil debe desplazarse surgen los diversos tipos de sistemas para la navegación autónoma. Sin embargo, existen cinco elementos que necesariamente estarán presentes en todo sistema de navegación autónoma:

- 1) Marco de referencia de posicionamiento.
- 2) Mecanismo para posicionamiento.
- 3) Sistema de Sensores del entorno.
- 4) Unidad de control / procesamiento.
- 5) Subsistemas para desplazamiento.

Por ejemplo, cuando un robot debe recorrer el espacio de una fábrica donde es utilizado, puede que su marco de referencia consista en un conjunto de marcas colocadas en el piso o las paredes de la fábrica por donde el robot se desplazará. Estas marcas pueden ser de varios tipos: placas metálicas, señales luminosas, magnéticas, etc. El robot debe ser capaz de detectar y reconocer esas marcas para poder guiarse a través del espacio de la fábrica. El conjunto de marcas o señalamientos constituyen el marco de referencia de posicionamiento del robot. Los medios por los cuales el robot consigue reconocer ese marco de referencia forman el mecanismo para posicionamiento del robot.

En los capítulos anteriores se han presentado cuatro de los elementos necesarios para la navegación autónoma empleados en este proyecto:

- Mecanismo para Posicionamiento (Capítulo 2, "El Sistema GPS").
- La Unidad de control / procesamiento (Capítulo 3, "El Microprocesador M68HC11").
- Sistema de Sensores del Entorno (Cap. 4, Sección 4.1 "Sensores").
- Subsistemas de Desplazamiento (Cap. 4, Sección 4.2 "Sistemas de Propulsión y Dirección").

En la sección 5.1 se presenta el último elemento necesario para la navegación autónoma: el Marco de Referencia de Posicionamiento. Se destina el resto del capítulo a la exposición del algoritmo de navegación diseñado para este proyecto. El algoritmo de navegación es la parte medular del sistema, pues es el elemento lógico o de decisión sobre el cual está escrito el programa del microprocesador que controla al resto de los elementos del sistema.



## 5.1 POSICIONAMIENTO

Las siglas WGS-84 designan el sistema coordinado de posicionamiento materializado y diseminado por la agencia norteamericana *National Imagery and Mapping Agency (NIMA)*. El origen de este sistema de referencia se remonta a la era Doppler, aunque en la actualidad prácticamente está basado en observaciones GPS. La versión más reciente es el denominado WGS-84 versión G873, de 1997 donde la letra "G" denota que la versión sólo contiene observaciones GPS. El número 873 indica la semana GPS en que las efemérides precisas calculadas por la NIMA se distribuyeron por primera vez al público en este nuevo sistema coordinado (0h UTC, Septiembre 29, 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 en un punto cercano a la intersección entre el Meridiano de Greenwich y el Eje Ecuatorial, lo cual da origen a los conocidos cuatro cuadrantes utilizados para posicionamiento. La figura 5.1 muestra un mapa referido al sistema WGS-84.

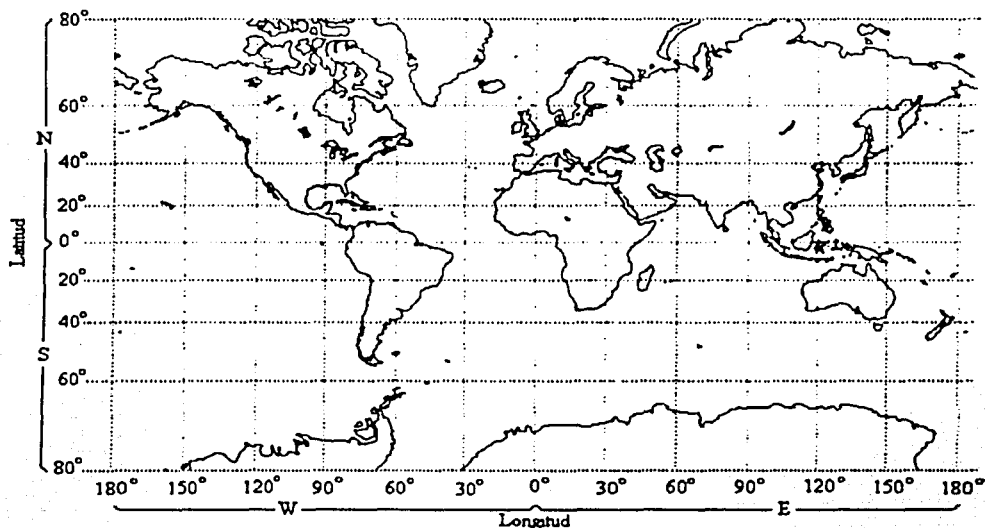
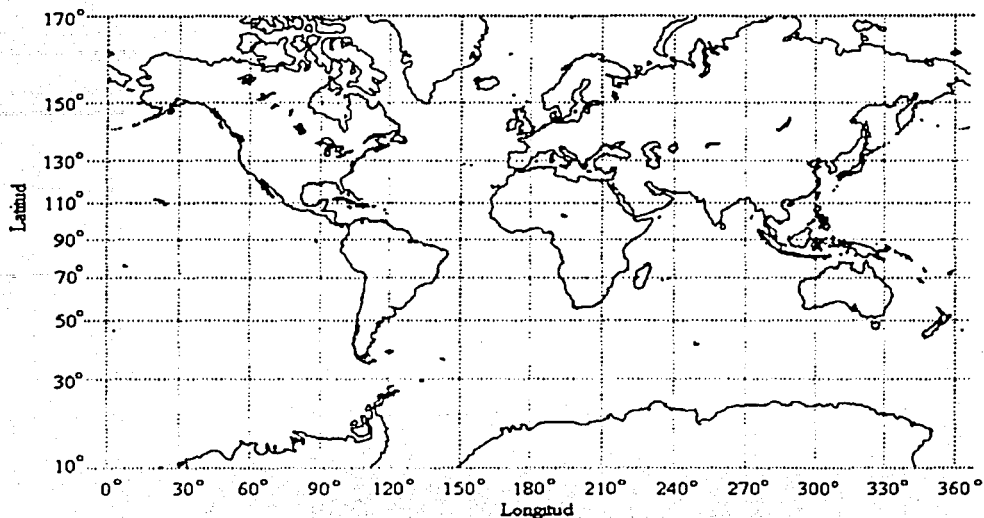


Figura 5.1 Mapa Global Referido al Sistema WGS-84

En el sistema 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, más un carácter asociado a cada valor Norte o Sur y Este u Oeste respectivamente. Sin embargo, para la manipulación de los datos de posición necesaria en el sistema de navegación presentado, se consideró más conveniente utilizar un marco de referencia distinto, en el cual los valores de latitud y longitud fueran absolutos. Esto se logra mediante la traslación del punto de origen del sistema WGS-84 a la posición (90° S, 180° W), con lo que todos los puntos de la superficie terrestre se ubican en un solo cuadrante. La figura 5.2 muestra un mapa en que se observa esta operación.



Al cambiar el origen del sistema de referencia ya no es necesario dividir en zonas el mapa. Las coordenadas son absolutas.

**Figura 5.2 Mapa Global Referido al Sistema Absoluto**

La traslación del origen del sistema WGS-84 al sistema absoluto resulta en las operaciones aplicadas a las coordenadas terrestres que muestra la tabla 5.1.

| Latitud (l) | Coordenada Absoluta (Y) | Longitud (L) | Coordenada Absoluta (X) |
|-------------|-------------------------|--------------|-------------------------|
| Norte       | $l + 90^\circ$          | Este         | $L + 180^\circ$         |
| Sur         | $90^\circ - l$          | Oeste        | $180^\circ - L$         |

**Tabla 5.1 Operaciones Para Cambio de Sistema de Referencia**

Por ejemplo, las coordenadas de la Ciudad de México en WGS-84 son aproximadamente ( $19^\circ$  N,  $99^\circ$  W) mientras que para el marco de referencia absoluto son :  $19^\circ + 90^\circ = 109^\circ$  de latitud y  $180^\circ - 99^\circ = 81^\circ$  de longitud, es decir ( $109^\circ$ ,  $81^\circ$ ). Esto facilita las operaciones matemáticas y la programación del algoritmo de navegación realizado para este proyecto. Cabe mencionar que a pesar de haber cambiado los valores de las coordenadas terrestres, se conserva su designación de latitud y longitud a lo largo de este escrito.

## 5.2 PLANIFICACIÓN DE TRAYECTORIAS

El punto de partida para la navegación es el posicionamiento del móvil. El siguiente paso es la planeación o planificación de la o las trayectorias que el móvil debe seguir para llegar a su destino. Los métodos de planeación de trayectorias para móviles terrestres parten del siguiente escenario elemental: dado un robot R y un entorno que contiene un conjunto de obstáculos fijos, se debe encontrar un camino, desde una configuración inicial I hasta una configuración final F, tal que R pueda viajar libremente desde I hasta G sin colisionar con ninguno de los obstáculos. Una configuración consiste en un par (posición, orientación) para del robot R en un instante cualquiera.

Los métodos que existen para realizar la planificación se dividen en tres grandes categorías: métodos de descomposición en celdas; métodos de mapa de caminos (*road-map*); y métodos de campos potenciales [1]. Estos métodos pueden actuar bien en el 'espacio de trabajo' del robot, es decir, el espacio en el que están definidos el robot y los obstáculos; o en el 'espacio de configuración', que es el espacio que consiste en todas las posibles configuraciones del robot. La ventaja de este último es que cualquier configuración del robot se representa por un solo punto en este espacio y así el problema de planificar el movimiento del robot (es decir, con sus dimensiones) se reduce a planificarlo para sólo un punto.

**Descomposición en celdas.**- En este método se construye un camino en el espacio de configuración, dividiendo el espacio libre (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 grafo en el que celdas adyacentes están conectadas. En estas celdas el movimiento del robot es normalmente fácil de calcular. Así, el movimiento que se busca realizar entre la configuración inicial y la final puede calcularse determinando las celdas que contienen esas configuraciones, encontrando un camino entre estas celdas en el grafo y calculando el movimiento entre cada arista del camino en el grafo.

**Mapa de caminos.**- En el método de mapa de caminos el problema básico del planificador consiste en que dados el punto inicial y el punto final, así como un 'Mapa de Nodos y Conexiones' que representan el entorno del robot, se debe planificar una trayectoria. Los nodos son puntos en los cuales puede estar el robot sin superponerse con algún objeto de su espacio de trabajo. Tanto el punto inicial como el punto final son considerados nodos del mapa. Sólo hay conexión entre dos nodos si no hay ningún objeto que cruce la línea recta que los une, es decir, las conexiones son caminos por los cuales es seguro viajar entre dos obstáculos. Debido a esta característica, al mapa de nodos y conexiones se le conoce como 'Mapa de Caminos', y de manera formal se le llama 'grafo de visibilidad'. Una vez que se tiene el grafo disponible, el movimiento se calcula desplazando el 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 hasta la configuración final.

El método de Descomposición en celdas y el de Mapas de caminos son llamados **métodos globales**: primero construyen una estructura de datos que luego 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 luego puede ser usada para muchas configuraciones inicial - final distintas. Sin embargo, estas estructuras de datos tienden a ser muy grandes, y los cálculos geométricos requeridos son a menudo difíciles y computacionalmente costosos.

**Campos Potenciales.**- Los métodos de campos potenciales son conocidos también como **métodos locales** y presentan una aproximación al problema de planificación radicalmente diferente a la de los métodos globales. En estos métodos sólo se calcula el movimiento para una configuración inicial-final dada. El método comienza en la configuración inicial e intentan mover el robot en pequeños pasos hacia el punto de destino. La dirección de estos pasos es determinada por una 'fuerza'. Se asume que el punto de destino produce una 'fuerza de atracción' en el robot; y los obstáculos en la trayectoria producen sobre él una 'fuerza repulsiva'. De esta forma el robot es 'atraído' hacia el objetivo o punto de destino mientras a su vez es 'repelido' por los obstáculos y así logra evitar 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 sí y el robot se detiene sin alcanzar su destino.

### 5.3 ALGORITMO DE NAVEGACIÓN PARA UN VEHÍCULO MARÍTIMO

Para el desarrollo del algoritmo de navegación del sistema propuesto, primero se intentó adaptar alguno de los métodos de planificación de trayectorias descritos en la sección 5.2. En poco tiempo fue claro que los métodos globales no son aplicables eficientemente a este sistema específico debido a que consideran condiciones iniciales que son diferentes a las que se plantean en este proyecto. Los métodos globales parten de la consideración de que el espacio de configuración del robot móvil puede modelarse en una estructura de datos con base en el conjunto de configuraciones del móvil y de las posiciones de los obstáculos en ese espacio. Esto presupone que las configuraciones del móvil son suficientemente estables como para configurar la estructura, es decir, que la posición del móvil y su orientación mantendrán un valor determinado de manera precisa mientras el robot por sí mismo no produzca los comandos dirigidos a cambiar esos valores. En otras palabras, la estructura de datos que modela el espacio de trabajo del robot no considera la posibilidad de perturbaciones en el medio físico que conforma dicho espacio.

En el caso de un vehículo marítimo, el espacio o medio físico de desplazamiento presenta por principio perturbaciones de diferentes tipos que modifican la configuración del móvil continuamente y sin un patrón definido (corrientes marinas, oleaje, viento, etc.). Esto prácticamente imposibilita la aplicación de los métodos globales de planificación, que dependen de la estabilidad del móvil. Adicionalmente, en este proyecto se toma en cuenta la posibilidad de que en la trayectoria del vehículo se presenten obstáculos dinámicos (aunque en un principio no fueron asentadas las características específicas de los obstáculos, éstas podían inferirse fácilmente de las características del medio) que en el planteamiento de los métodos globales por lo general tampoco aparecen.

También se estudió el método de campos potenciales, encontrando que es un método que permite un buen grado de distorsión en la configuración del móvil pues continuamente procesa datos actualizados de la posición y la orientación del mismo para realizar ajustes en la trayectoria de desplazamiento. Se optó entonces por desarrollar un algoritmo del tipo de campos potenciales. Sin embargo, fue evidente que la mayor virtud de este método se desperdicia cuando el espacio de trabajo del móvil carece de una considerable concentración de obstáculos, puesto que entonces desaparece la mayoría de las 'fuerzas' o 'potenciales' que participan en el algoritmo de control de navegación. Además, tras un análisis más detallado, se observa que la implementación de un sistema de navegación basado en ese método requiere de un sistema de sensores más complejo del que se contempla para el sistema propuesto.

Finalmente, se concluyó que si bien no era posible desarrollar el algoritmo de control de navegación basándose estrictamente en el método de campos potenciales, el método de planificación de trayectorias más adecuado para el sistema de navegación debía tener un enfoque local y seguir un análisis de la 'cinemática inversa' del móvil tal como el método de campos potenciales. En términos generales, en la planeación global el sistema robótico es fijo, mientras que en la planificación local lo fijo es el ambiente. Al diseñar el algoritmo de control se consideró que el móvil mantendría un movimiento constante, por lo que el esfuerzo se concentró en el cálculo de la corrección de su trayectoria, con el fin de que se dirija a su destino.

El análisis del movimiento de un robot móvil puede realizarse bajo dos criterios: cuando se calcula el movimiento de las variables del sistema para conocer el punto de destino final del mismo, se trata de un análisis de la 'cinemática directa' del sistema; en caso de que los movimientos de los elementos del sistema se deriven de la posición final que se pretende alcanzar, el análisis se conoce como de 'cinemática inversa'. De cierto modo, el uso de la cinemática inversa se desprende de manera natural de las características de los vehículos marítimos que se desean controlar, pues se considera que el motor de dirección del móvil es independiente del motor de propulsión.

Como el motor que indica el ángulo que debe girar el móvil para lograr una orientación no interfiera con el motor que proporciona la fuerza para su desplazamiento, el procedimiento más simple para el control del vehículo consiste precisamente en encontrar el ángulo de giro necesario para que el móvil alcance la coordenada deseada, es decir, trabajando con cinemática inversa.

El resto del capítulo se dedica a la explicación del algoritmo de navegación marítima que se diseñó. Puede observarse que en un sentido estricto no se trata de un algoritmo de navegación, pues se parte del supuesto de que el espacio de configuraciones del móvil es prácticamente espacio libre. A diferencia de los métodos de planificación de trayectorias expuestos, que buscan encontrar la mejor solución a la ruta entre las configuraciones inicial y final del móvil, este algoritmo de control únicamente garantiza que el móvil encontrará una ruta hacia su destino, y no tiene la capacidad de realizar una optimización de la trayectoria.

Aunque dentro del planteamiento del problema de navegación del móvil se incluyó como parte esencial la evasión de obstáculos, esta se trata de forma separada del algoritmo de navegación, de modo que ante la presencia de un obstáculo en la trayectoria el sistema conmuta el control del móvil a una subrutina específica que no forma parte integral del algoritmo de navegación y que no considera la configuración del móvil para su ejecución. Por estas razones, el algoritmo de navegación puede clasificarse también como un algoritmo de posicionamiento; sin embargo, a lo largo de este escrito se mantiene el nombre de 'algoritmo de navegación' porque éste ilustra de una mejor manera la función esencial del algoritmo.

### 5.3.1 Estructura Del Algoritmo De Navegación Marítima

La primera parte del algoritmo consiste en posibilitar la determinación de la configuración actual del móvil. Puesto que se maneja un espacio bidimensional para el desplazamiento del mismo, sus configuraciones están dadas por tres variables: dos de traslación y una de rotación. Así, se representa la configuración del móvil por  $(X, Y, \gamma)$ , donde  $X$  y  $Y$  son las coordenadas del móvil en el sistema absoluto y  $\gamma$  es su ángulo de trayectoria. Se llama ángulo de trayectoria y al ángulo que se forma con la horizontal o eje de las longitudes y la recta que pasa por los puntos definidos por la posición instantánea y la posición inicial del móvil. Debe notarse que las coordenadas de la posición instantánea sustituyen a las de la posición inicial en cada lectura que se realiza de la señal GPS. La figura 5.3 ilustra esta operación.

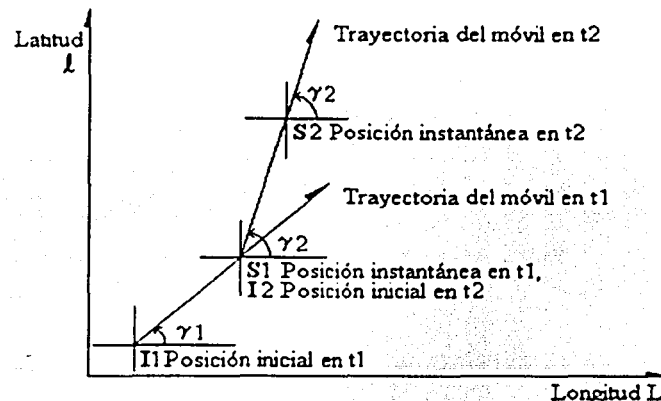


Figura 5.3 Actualización de Coordenadas del Móvil

En el instante  $t_1$  se toma una lectura de la señal GPS, cuando el móvil se encuentra en la posición  $S_1$ . Para ese instante  $t_1$  existe una posición inicial  $I_1$ , que es la posición del móvil en el momento de la lectura anterior de la señal GPS. El ángulo de trayectoria de este instante  $t_1$  se calcula con base en estas dos posiciones,  $\gamma_1$ . En un instante  $t_2$  se toma otra lectura GPS, con el móvil en la posición  $S_2$ . Para  $t_2$  existe una posición inicial  $I_2$ , que es la posición del móvil en el momento de la lectura anterior del GPS, esto es,  $S_1$ . Para  $t_2$  se calcula otro ángulo de trayectoria  $\gamma_2$ , con base en las posiciones  $S_2$  e  $I_2$ .

Se desarrolló el proceso de actualización de la configuración del móvil porque se considera que la trayectoria del móvil ha de cambiar entre lecturas por causas diversas, como la presencia de un obstáculo, una desviación provocada por el medio, etc. Se debió tomar en cuenta que para la obtención de una configuración del móvil son necesarias dos lecturas del receptor GPS en posiciones distintas, pues el sistema GPS no aporta información acerca de la orientación del móvil. Por tanto, en la primera lectura de la señal GPS no puede calcularse el ángulo  $\gamma$ , pues no existe lectura anterior en ese instante, de modo que se espera una segunda lectura para calcular la trayectoria del móvil y su configuración por primera vez.

Para el cálculo del ángulo de trayectoria  $\gamma$ , se procede de la siguiente manera: primero se obtienen las diferencias entre las longitudes ( $\Delta L$ ) y latitudes ( $\Delta l$ ) de las posiciones instantánea e inicial, luego se realiza el cociente de la diferencia de latitudes entre la diferencia de longitudes y así se obtiene la tangente del ángulo buscado ( $\tan \gamma = \Delta l / \Delta L$ ). Enseguida se compara el valor encontrado con los valores de la tabla de tangentes previamente creada y así se obtiene el ángulo deseado con un rango de precisión de tres grados a partir del valor de tangente calculado (refiérase al capítulo 6, sección 6.3.2 para una explicación más detallada). La ubicación de la posición instantánea respecto de la posición inicial puede presentarse en cuatro casos, esto afectará los cálculos realizados posteriormente. La figura 5.4 ilustra estos cuatro casos para el cálculo del ángulo de trayectoria.

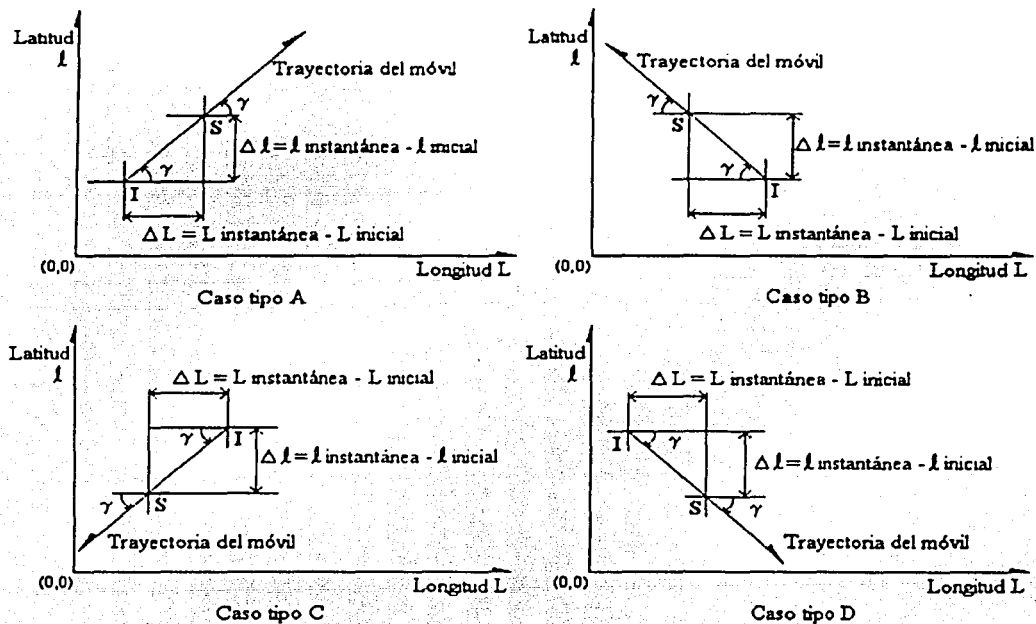


Figura 5.4 Posición Relativa: Posición Instantánea Respecto a Inicial Para el Ángulo  $\gamma$

La tabla 5.2 condensa las condiciones que se aplican a cada caso posible para el cálculo del ángulo de trayectoria.

| Caso    | Diferencia de longitudes | Diferencia de latitudes | Angulo $\gamma$ |
|---------|--------------------------|-------------------------|-----------------|
| Tipo A: | $\Delta L = (+)$         | $\Delta \ell = (+)$     | $\gamma = (+)$  |
| Tipo B: | $\Delta L = (-)$         | $\Delta \ell = (+)$     | $\gamma = (-)$  |
| Tipo C: | $\Delta L = (-)$         | $\Delta \ell = (-)$     | $\gamma = (-)$  |
| Tipo D: | $\Delta L = (+)$         | $\Delta \ell = (-)$     | $\gamma = (+)$  |

Tabla 5.2 Condiciones Para los Distintos Casos en el Cálculo del Ángulo  $\gamma$

Una vez calculado el ángulo de trayectoria  $\gamma$ , se guardan los signos de las diferencias de longitud  $\Delta L$  y de latitud  $\Delta \ell$  y se procede a llenar un registro de control con la información de cual fue el tipo de caso que se presentó. Esta operación se realiza de acuerdo al diagrama de flujo que muestra la figura 5.5. El registro de control se utiliza posteriormente dentro del algoritmo de navegación en el cálculo del ángulo de corrección  $\beta$ .

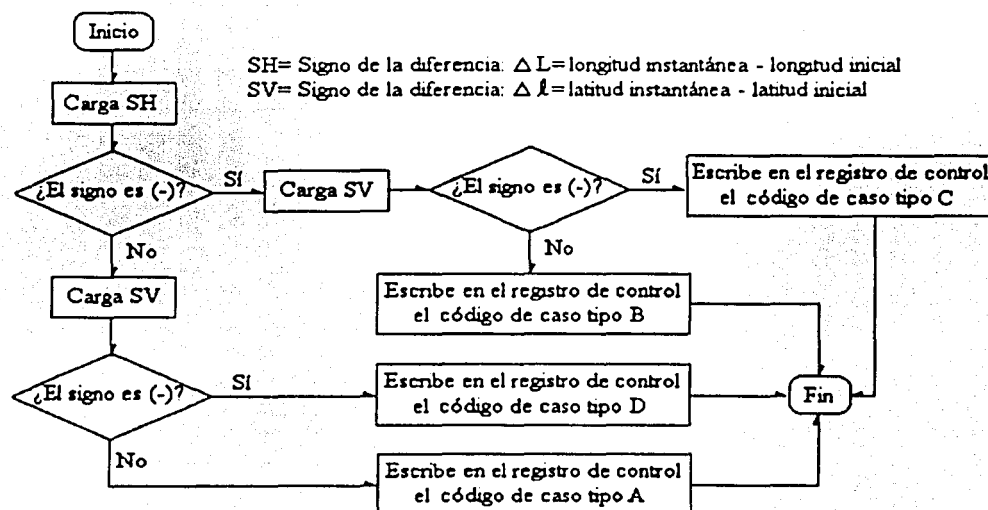


Figura 5.5 Diagrama de Flujo Para el Llenado del Registro de Control

La segunda parte del algoritmo consiste en calcular otro ángulo, llamado ángulo de dirección  $\alpha$ , y que es el ángulo que se forma entre la horizontal o eje de las longitudes, con la recta que pasa por los puntos definidos por las coordenadas de la posición final o de destino, y la posición actual o instantánea del móvil en cada lectura de la señal GPS.

El ángulo de dirección  $\alpha$  se obtiene de forma muy parecida al ángulo de trayectoria  $\gamma$ , la única diferencia consiste en que para su cálculo se utilizan las coordenadas de la posición final en vez de las de la posición instantánea y las coordenadas de la posición instantánea en vez de las de la posición inicial. Al igual que para  $\gamma$ , hay cuatro casos que pueden presentarse al momento de calcular  $\alpha$ , dependiendo de la ubicación de la posición final respecto de la instantánea. La tabla 5.3 resume las condiciones de estos cuatro casos.

| Caso    | Diferencia de longitudes | Diferencia de latitudes | Angulo $\alpha$ |
|---------|--------------------------|-------------------------|-----------------|
| Tipo 1: | $\Delta L = (+)$         | $\Delta \ell = (+)$     | $\alpha = (+)$  |
| Tipo 2: | $\Delta L = (-)$         | $\Delta \ell = (+)$     | $\alpha = (-)$  |
| Tipo 3: | $\Delta L = (-)$         | $\Delta \ell = (-)$     | $\alpha = (-)$  |
| Tipo 4: | $\Delta L = (+)$         | $\Delta \ell = (-)$     | $\alpha = (+)$  |

Tabla 5.3 Condiciones para los distintos casos en el cálculo del ángulo  $\alpha$

Asimismo, debe llenarse de nuevo el registro de control, para que indique el caso que se presentó en el cálculo del ángulo  $\alpha$ , el proceso para llenar el registro de control con el tipo de caso para  $\alpha$  es el mismo que se utilizó para  $\gamma$ , únicamente cambia el código que se escribe en el registro de control. Cuando se registra el tipo de caso para  $\alpha$ , no se afecta la parte del registro previamente escrita, de modo que el registro de control final refleja una combinación entre el caso de  $\gamma$  y el caso de  $\alpha$ .

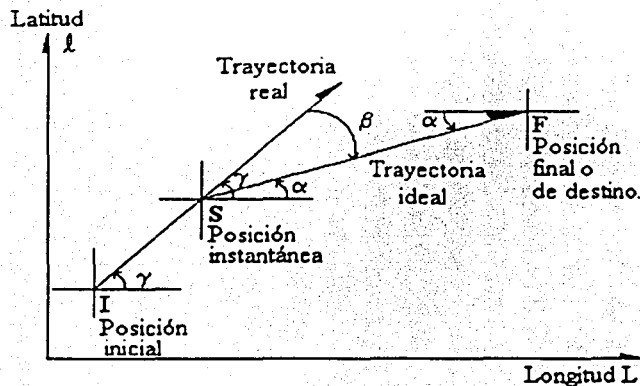


Figura 5.6 El Ángulo de Trayectoria Gamma, El ángulo de Dirección Alfa y el Ángulo de Corrección Beta.

En la tercera parte del algoritmo, se utilizan los dos ángulos ya calculados para obtener un tercer ángulo, el ángulo de corrección  $\beta$ . Éste es el ángulo que girará el motor de dirección del móvil, con el fin de corregir la trayectoria del mismo y de que se dirija a su destino. El cálculo de  $\beta$  depende de los casos que se hayan presentado en los cálculos de  $\alpha$  y de  $\gamma$ . Para ilustrar las distintas posibilidades se presentan las figuras 5.7, 5.8, 5.9 y 5.10 obtenidas al combinar los casos correspondientes tanto para el cálculo de  $\gamma$  como el de  $\alpha$ .



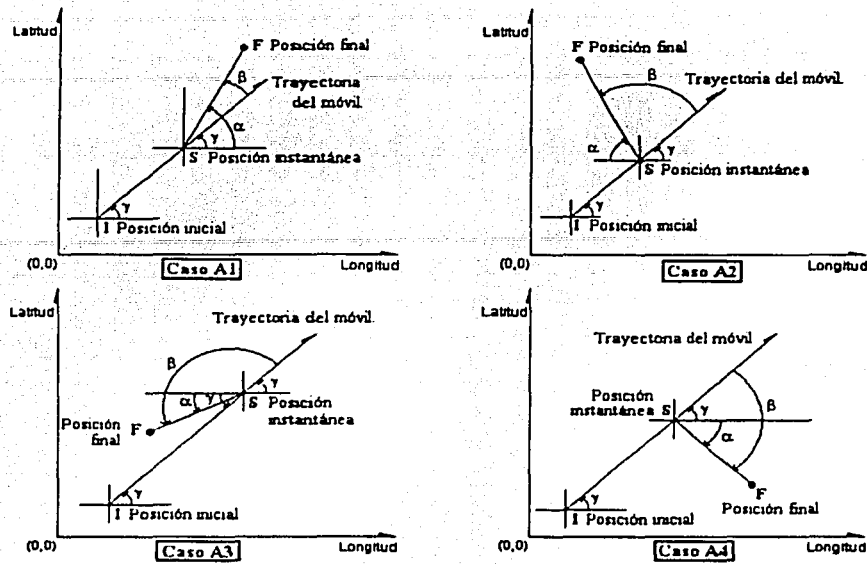


Figura 5.7 Casos de Orientación Tipo Ax.

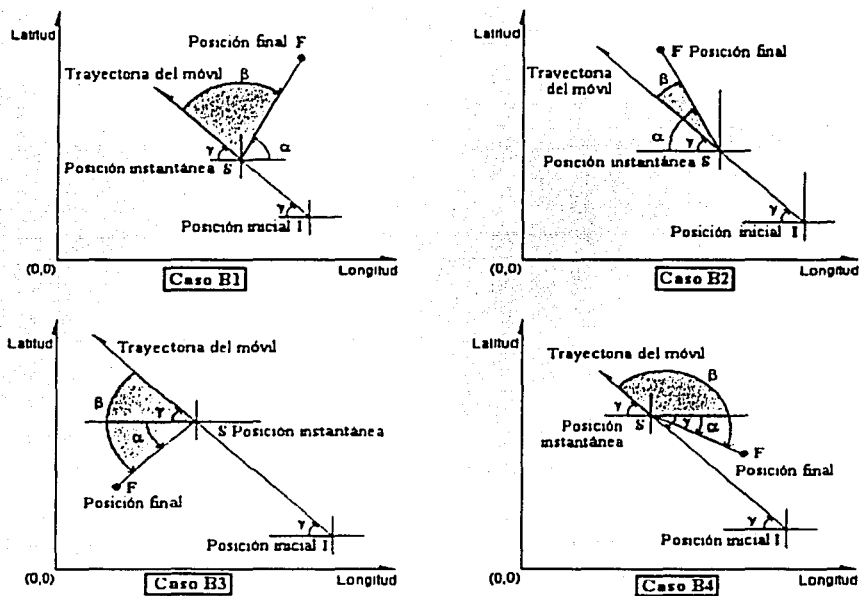


Figura 5.8 Casos de Orientación Tipo Bx.

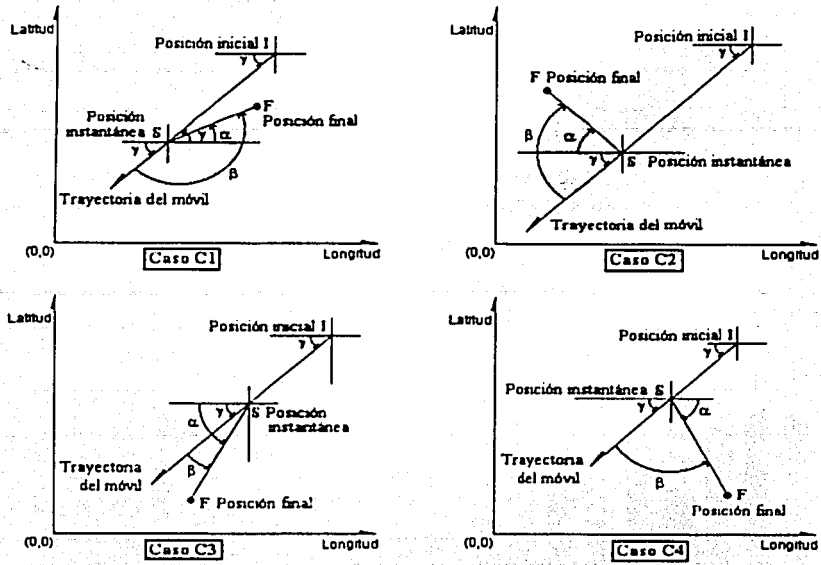


Figura 5.9 Casos de Orientación Tipo Cx.

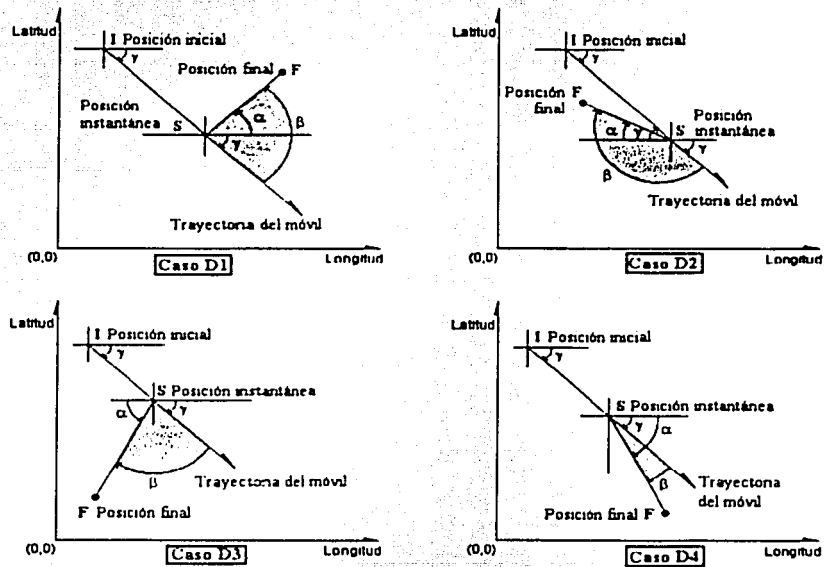


Figura 5.10 Casos de Orientación Tipo Dx

A continuación, la tabla 5.4 concentra todas las expresiones matemáticas que definen los valores del ángulo beta; también se muestran las condiciones de las diferencias de latitud y longitud, el signo del ángulo gamma y ángulo alfa en cada caso, los valores límites que puede tomar el ángulo beta y el sentido de giro que realizará el motor de dirección.

| Caso                 | L l $\gamma$ | L l $\alpha$ | Expresión para $ \beta $                      | Rango de $ \beta $ | Sentido |
|----------------------|--------------|--------------|---|--------------------|---------|
| A1 $\alpha > \gamma$ | +++          | +++          | $ \beta  = (\alpha - \gamma)$                 | 0° a 90°           | Der (-) |
| A1 $\gamma > \alpha$ | +++          | +++          | $ \beta  = (\gamma - \alpha)$                 | 0° a 90°           | Izq (+) |
| A2 $\alpha > \gamma$ | +++          | -+-          | $ \beta  = 180^\circ - ( \alpha  +  \gamma )$ | 0° a 180°          | Der (-) |
| A2 $\gamma > \alpha$ | +++          | -+-          | $ \beta  = 180^\circ - ( \alpha  +  \gamma )$ | 0° a 180°          | Der (-) |
| A3 $\alpha > \gamma$ | +++          | --+          | $ \beta  = 180^\circ - (\alpha - \gamma)$     | 90° a 180°         | Izq (+) |
| A3 $\gamma > \alpha$ | +++          | --+          | $ \beta  = 180^\circ - (\gamma - \alpha)$     | 90° a 180°         | Der (-) |
| A4 $\alpha > \gamma$ | +++          | +- -         | $ \beta  = ( \alpha  +  \gamma )$             | 0° a 180°          | Izq (+) |
| A4 $\gamma > \alpha$ | +++          | +- -         | $ \beta  = ( \alpha  +  \gamma )$             | 0° a 180°          | Izq (+) |
| B1 $\alpha > \gamma$ | -+-          | +++          | $ \beta  = 180^\circ - ( \alpha  +  \gamma )$ | 0° a 180°          | Izq (+) |
| B1 $\gamma > \alpha$ | -+-          | +++          | $ \beta  = 180^\circ - ( \alpha  +  \gamma )$ | 0° a 180°          | Izq (+) |
| B2 $\alpha > \gamma$ | -+-          | -+-          | $ \beta  = ( \alpha  -  \gamma )$             | 0° a 90°           | Izq (+) |
| B2 $\gamma > \alpha$ | -+-          | -+-          | $ \beta  = ( \gamma  -  \alpha )$             | 0° a 90°           | Der (-) |
| B3 $\alpha > \gamma$ | -+-          | --+          | $ \beta  = ( \alpha  +  \gamma )$             | 0° a 180°          | Der (-) |
| B3 $\gamma > \alpha$ | -+-          | --+          | $ \beta  = ( \alpha  +  \gamma )$             | 0° a 180°          | Der (-) |
| B4 $\alpha > \gamma$ | -+-          | +- -         | $ \beta  = 180^\circ - ( \alpha  -  \gamma )$ | 90° a 180°         | Der (-) |
| B4 $\gamma > \alpha$ | -+-          | +- -         | $ \beta  = 180^\circ - ( \gamma  -  \alpha )$ | 90° a 180°         | Izq (+) |
| C1 $\alpha > \gamma$ | --+          | +++          | $ \beta  = 180^\circ - (\alpha - \gamma)$     | 90° a 180°         | Izq (+) |
| C1 $\gamma > \alpha$ | --+          | +++          | $ \beta  = 180^\circ - (\gamma - \alpha)$     | 90° a 180°         | Der (-) |
| C2 $\alpha > \gamma$ | --+          | -+-          | $ \beta  = ( \alpha  +  \gamma )$             | 0° a 180°          | Izq (+) |
| C2 $\gamma > \alpha$ | --+          | -+-          | $ \beta  = ( \alpha  +  \gamma )$             | 0° a 180°          | Izq (+) |
| C3 $\alpha > \gamma$ | --+          | --+          | $ \beta  = (\alpha - \gamma)$                 | 0° a 90°           | Der (-) |
| C3 $\gamma > \alpha$ | --+          | --+          | $ \beta  = (\gamma - \alpha)$                 | 0° a 90°           | Izq (+) |
| C4 $\alpha > \gamma$ | --+          | +- -         | $ \beta  = 180^\circ - ( \alpha  +  \gamma )$ | 0° a 180°          | Der (-) |
| C4 $\gamma > \alpha$ | --+          | +- -         | $ \beta  = 180^\circ - ( \alpha  +  \gamma )$ | 0° a 180°          | Der (-) |
| D1 $\alpha > \gamma$ | +- -         | +++          | $ \beta  = ( \alpha  +  \gamma )$             | 0° a 180°          | Der (-) |
| D1 $\gamma > \alpha$ | +- -         | +++          | $ \beta  = ( \alpha  +  \gamma )$             | 0° a 180°          | Der (-) |
| D2 $\alpha > \gamma$ | +- -         | -+-          | $ \beta  = 180^\circ - ( \alpha  -  \gamma )$ | 90° a 180°         | Der (-) |
| D2 $\gamma > \alpha$ | +- -         | -+-          | $ \beta  = 180^\circ - ( \gamma  -  \alpha )$ | 90° a 180°         | Izq (+) |
| D3 $\alpha > \gamma$ | +- -         | --+          | $ \beta  = 180^\circ - ( \alpha  +  \gamma )$ | 0° a 180°          | Izq (+) |
| D3 $\gamma > \alpha$ | +- -         | --+          | $ \beta  = 180^\circ - ( \alpha  +  \gamma )$ | 0° a 180°          | Izq (+) |
| D4 $\alpha > \gamma$ | +- -         | +- -         | $ \beta  = ( \alpha  -  \gamma )$             | 0° a 90°           | Izq (+) |
| D4 $\gamma > \alpha$ | +- -         | +- -         | $ \beta  = ( \gamma  -  \alpha )$             | 0° a 90°           | Der (-) |

Tabla 5.4 Condiciones para obtener el ángulo Beta

Al analizar la tabla 5.4 se observa que en varios casos las expresiones matemáticas para calcular el ángulo beta y el sentido del movimiento del motor de dirección son idénticos, debido a que las posiciones relativas entre los puntos definidos coinciden. Por este motivo, pueden integrarse en una sola expresión los casos repetidos obteniendo la tabla 5.5.

| Caso  |                   | Beta                                      | Sentido |
|-------|-------------------|---|---------|
| A1,C3 | $\alpha > \gamma$ | $ \beta  = \alpha - \gamma$               | DER     |
|       | $\gamma > \alpha$ | $ \beta  = \gamma - \alpha$               | IZQ     |
| A2,C4 | ----              | $ \beta  = 180^\circ - (\alpha + \gamma)$ | DER     |
| A3,C1 | $\alpha > \gamma$ | $ \beta  = 90^\circ$                      | IZQ     |
|       | $\gamma > \alpha$ | $ \beta  = 90^\circ$                      | DER     |
| A4,C2 | ----              | $ \beta  = \alpha + \gamma$               | IZQ     |
| B1,D3 | ----              | $ \beta  = 180^\circ - (\alpha + \gamma)$ | IZQ     |
| B2,D4 | $\alpha > \gamma$ | $ \beta  = \alpha - \gamma$               | IZQ     |
|       | $\gamma > \alpha$ | $ \beta  = \gamma - \alpha$               | DER     |
| B3,D1 | ----              | $ \beta  = \alpha + \gamma$               | DER     |
| B4,D2 | $\alpha > \gamma$ | $ \beta  = 90^\circ$                      | DER     |
|       | $\gamma > \alpha$ | $ \beta  = 90^\circ$                      | IZQ     |

Tabla 5.5 Tabla Condensada de Condiciones Para Beta

En el algoritmo de navegación se lee el registro de control que contiene la combinación correspondiente, y a partir de la tabla anterior se calcula el ángulo beta. Nótese que la expresión para beta depende de las magnitudes relativas entre los ángulos gamma y alfa, y así también el sentido de giro del motor de dirección.

Siguiendo las correcciones calculadas el móvil alcanzará el punto de destino. Dentro del capítulo siguiente, de integración del sistema, se explican con detalle las especificaciones técnicas consideradas para el cálculo de cada ángulo mencionado a lo largo de este capítulo (rango de valores posibles y precisión, tiempo de muestreo de señal GPS, velocidad del móvil, etc.).

## REFERENCIAS BIBLIOGRÁFICAS

- [1] Guardiola, Eduardo; "Planificación de movimientos con rotaciones, Robótica"; 2000  
<http://www.dccia.ua.es/dccia/inf/asignaturas/ROB/optativos/Planificacion/>

## 6

## Integración del Sistema

En este capítulo se describe la integración, es decir, cómo todos los elementos del sistema de control de navegación se conjuntan de una manera funcional. La explicación precedente se realiza desde dos enfoques simultáneos: se trata de presentar paso por paso el funcionamiento del programa principal escrito para el sistema a la vez que se describen las características técnicas e interfaces que restan por discutir para lograr la integración del mismo.

### 6.1. PROTOCOLO DE ENCENDIDO DEL SISTEMA

Durante la fase de diseño y experimentación del sistema, se utilizaron dos fuentes de alimentación para los distintos subsistemas: una batería recargable de ácido de 6 (V) y 10 (Ah) para el receptor GPS y el motor de dirección, y una batería recargable de 12 (V) y 10 (Ah) para el microprocesador y el motor de propulsión. Independientemente, se realizaron pruebas para el subsistema de sensores con una fuente de 12 (V). Con base en esta configuración se siguió el siguiente protocolo de encendido y puesta en marcha del sistema:

1. Encendido del receptor GPS y polarización del motor de dirección (Switch S1).
2. Tiempo de espera (50 segundos) para que se establezca el receptor GPS.
3. Encendido del microprocesador y polarización del motor de propulsión (Switch S2).
4. Inicialización del programa del microprocesador, introducción y confirmación de las coordenadas de destino a través de una PC.
5. Habilitación de la comunicación serial entre receptor GPS y microprocesador (Switch S3).

Como referencia rápida se muestra a continuación la tabla 6.1 que relaciona cada dispositivo empleado en el sistema con su fuente de alimentación así como con el interruptor que habilita cada fuente. Se ha incluido el sistema de sensores ultrasónicos. Los interruptores o *switches* marcados S1 y S2 son interruptores manuales, el interruptor C1 para los sensores es el circuito integrado 4016 (figura 4.9) controlado por el microprocesador.

| Dispositivo           | Fuente de alimentación (V) | Switch de habilitación |
|-----------------------|----------------------------|------------------------|
| Microprocesador       | 12                         | S2                     |
| Motor de propulsión   | 12                         | S2                     |
| Sensores ultrasónicos | 12                         | C1                     |
| Motor de dirección    | 6                          | S1                     |
| Receptor GPS          | 6                          | S1                     |

Tabla 6.1 Fuentes de Alimentación e Interruptores de Habilitación del Sistema

Una vez que se han cumplido los cinco pasos de este protocolo de encendido, y hasta que el móvil alcance su destino, el sistema es completamente autónomo. Puede notarse, sin embargo, que durante la puesta en marcha del sistema, la intervención del usuario es mínima, y que ciertamente el paso crítico del protocolo de encendido es la inicialización del programa del microprocesador y la introducción de las coordenadas de destino para el móvil. En la explicación siguiente, por tanto, se tratan con detalle estos procedimientos.



- 1) Ocho caracteres para el valor de latitud: un entero de dos dígitos con el valor de los grados de latitud, punto decimal y cinco caracteres indicando las fracciones de grado de latitud.
- 2) Un carácter, "N" o "S", que indica la orientación de la latitud.
- 3) Ocho caracteres para el valor de longitud: un entero de uno, dos o tres dígitos con el valor de los grados de longitud, punto decimal y seis, cinco o cuatro caracteres respectivamente, indicando las fracciones de grado de longitud.
- 4) Un espacio en blanco, seguido de un carácter "E" o "W" que indica la orientación de la longitud.

Como ejemplo, si se analiza la figura 6.1, se observa la introducción de las coordenadas de destino:

19.32946N99.18031 W

Lo anterior corresponde a: 19 grados de latitud; 3 décimas, 2 centésimas, 9 milésimas, 4 diezmilésimas, y 6 cienmilésimas de grado de latitud; "N" indicando que la latitud es norte; 99 grados de longitud; 1 décima, 8 centésimas, 0 milésimas, 3 diezmilésimas, 8 y 1 cienmilésima de grado de longitud; un espacio y "W" que indica que la longitud es oeste.

El formato anterior se introdujo con el fin de facilitar al programa la identificación de los valores. El carácter "N" o "S" para el caso de la latitud no sólo indica la orientación de la coordenada correspondiente, sino que sirve de marca al programa para identificar la terminación del valor introducido. En el caso de la longitud es necesario incluir además del carácter "E" o "W" un espacio como marca, pues de lo contrario el programa interpretaría erróneamente que el carácter "E" precede a un exponente (por ejemplo, 9.9E04 significa  $9.9 \times 10^4$ ), lo que ocasionaría la incorrecta ejecución del programa.

Una vez introducidas las coordenadas del destino, éstas se despliegan para su confirmación. El programa esperará la señal de confirmación, el carácter "S" si son correctas o "N" si son incorrectas; en caso de ser incorrectas, el programa no tomará en cuenta tales coordenadas y esperará a que se introduzcan unas nuevas. Esta tarea se repetirá hasta que se hayan confirmado como correctas las coordenadas introducidas. Posteriormente, se podrá desconectar la PC del microprocesador con el fin de dar paso a las siguientes etapas del programa que serán ejecutadas consecutivamente y de manera completamente autónoma.

## 6.3 AUTONOMÍA DEL SISTEMA

### 6.3.1 Toma De Lecturas Del Receptor GPS

Únicamente después de confirmar las coordenadas del destino se deberá encender el switch S3 que habilita la comunicación serial con el receptor GPS. Como segunda etapa, se realiza la primera lectura del receptor GPS, que determina la posición instantánea del móvil y se manda un tren de pulsos con un ciclo de trabajo del 80% al motor de propulsión para que avance el móvil.

El receptor GPS entrega varios mensajes, cada uno de ellos con información dirigida a diferentes aplicaciones. En este proyecto se utiliza el mensaje GPGGA para navegación marítima. Para poder usar este mensaje primero fue necesario identificarlo y separarlo del resto de los mensajes obtenidos del receptor GPS. Para ello se desarrolló un algoritmo que consiste en la comparación de los encabezados de todas las mensajes recibidas con el encabezado "GPGGA" y la verificación de que la trama del mensaje esté completa.

Después de identificar el mensaje GPGGA y con base en la posición definida en la norma NMEA-083, se extraen los valores de las coordenadas terrestres generando una cadena de caracteres en un formato conveniente para realizar operaciones mediante las rutinas de punto flotante de Motorola [1], obteniendo los valores de las coordenadas terrestres en unidades de grados y fracciones de grado, eliminando la notación de grados y minutos. Finalmente, los valores se transforman al sistema de referencia absoluto de modo que pueden ser empleados en cualquier parte del algoritmo de navegación.

### 6.3.2 Rastreo De Posición y Corrección De Trayectoria

La tercera etapa del programa se encarga de verificar si el móvil llegó a su destino, para esto se definió un límite de acercamiento dentro del cual se considera que el móvil ha alcanzado su destino. Este límite se puede ajustar siempre que no resulte menor a la precisión propia del GPS (valor mínimo nominal de 8 (m) aproximadamente). El procedimiento para obtener el límite de acercamiento consiste en la comparación de las coordenadas de destino con las coordenadas instantáneas (tanto latitudes como longitudes). Cuando la diferencia entre ambas coordenadas sea menor o igual al límite, el motor de propulsión se detiene y el programa finaliza. Mientras la diferencia sea mayor que el límite establecido, el programa continúa con la siguiente etapa. El valor actual del límite de acercamiento es de 0.00011 grados, equivalente a 12 (m) aproximadamente. Este valor se derivó de las características consideradas para el móvil.<sup>4</sup>

La siguiente etapa es el cálculo de las correcciones de trayectoria a partir de los datos obtenidos con el receptor GPS, calculando los ángulos gamma, alfa y beta según el algoritmo expuesto en el capítulo 5. Es importante recordar que para el cálculo de los ángulos alfa y gamma se recurre a una tabla de valores de tangentes de ángulos contra la cual se compara la tangente del ángulo que se quiere calcular. Al ocurrir una comparación exitosa se obtiene el ángulo buscado.

Se utiliza este procedimiento para la obtención de los ángulos debido a que es mucho más sencillo programar la comparación entre un valor calculado y una tabla de valores fijos que programar la función de tangente inversa o 'ángulo cuya tangente'. Sin embargo, esto necesariamente conlleva una pérdida en la precisión del cálculo puesto que en una tabla el número de valores es limitado, mientras que con una función se tiene un número infinito de resultados posibles. La tabla de valores de tangentes que se utiliza contiene treinta valores definidos de tangente, esto corresponde a un rango de 0° a 90° dividido en 30 intervalos de 3° cada uno.

Cuando se calcula el ángulo alfa o el ángulo gamma, primero se obtiene la tangente de ese ángulo, luego se compara este valor con los valores definidos en la tabla de tangentes, comenzando por la tangente de cero grados e incrementándose progresivamente. Si el valor de la tangente del ángulo buscado resulta ser mayor que el primer valor de la tabla de tangentes, se realizará la comparación con el siguiente valor en la tabla. La comparación continúa de esta manera hasta que el valor de la tangente del ángulo buscado no sea mayor que uno de los valores de la tabla de tangentes, lo que significa una comparación 'Negativa'. Entonces al ángulo buscado se le asigna el valor cuya tangente arrojó esa comparación Negativa. La explicación anterior se comprende mejor con un ejemplo:

Ejemplo para la obtención del ángulo gamma o del ángulo alfa.

- 1) Se supone que la tangente del ángulo buscado es: 0.624869 este valor se obtiene siguiendo el procedimiento descrito en el Capítulo 5 (5.2 Algoritmo de Navegación) según sea el caso para alfa o gamma.

<sup>4</sup> El móvil que se pretende controlar es un vehículo marítimo de aproximadamente 2 (m) o 2.5 (m) de envergadura, con una velocidad de 5 (m/s) o 18 (Km/Hr).



- 2) Se comienza a comparar el valor 0.624869 con los valores de la tabla de tangentes (el primer valor es 0, pues  $\tan(0^\circ) = 0$ ). Así, la primera comparación es: ¿  $0 < 0.624869$  ? La comparación es Positiva.
- 3) Se prosigue la comparación con el siguiente valor ( $0.052407 = \tan(3^\circ)$ ) la segunda comparación es: ¿  $0.052407 < 0.624869$  ? La comparación es Positiva nuevamente.
- 4) La comparación continúa con los valores de las tangentes de  $6^\circ$ ,  $9^\circ$ ,  $12^\circ$ , etc. y luego con el valor de la tangente de  $30^\circ$ ;  $\tan(30^\circ) = 0.577350$   
Así que se compara: ¿  $0.577350 < 0.624869$  ? La comparación también es Positiva.
- 5) La siguiente comparación es con el valor de la tangente de  $33^\circ$ ;  $\tan(33^\circ) = 0.649407$   
Se compara: ¿  $0.649407 < 0.624869$  ? La respuesta es Negativa, por lo tanto, según este procedimiento el ángulo buscado es  $33^\circ$ .

En el ejemplo anterior, el ángulo cuya tangente es 0.624869 es  $32^\circ$ , por lo que hubo un error en el cálculo del ángulo de  $1^\circ$ , dado que se identificó el valor del ángulo como  $33^\circ$ . La mayor diferencia que puede existir entre el ángulo cuya tangente es un determinado valor y el ángulo que se obtiene por la comparación de ese valor con la tabla de tangentes es de  $3^\circ$  pues es el valor definido de cada intervalo de la tabla.

Esta diferencia entre el valor real del ángulo y el valor obtenido por comparación con la tabla es aceptable para los fines que se persiguen en este proyecto, ya que el algoritmo de navegación se diseñó pensando en que se generen correcciones de la trayectoria del móvil de manera relativamente continua, por lo que una desviación del orden de  $6^\circ$  (considerando que se sume el máximo error en la obtención del ángulo alfa con el máximo error en la obtención del ángulo gamma) afecta de forma mínima la trayectoria recorrida por el móvil. Además, debe mencionarse que la precisión del método descrito para la obtención de los ángulos puede incrementarse fácilmente si así se requiere, bastará definir una mayor cantidad de valores en la tabla de tangentes utilizada para la comparación.

Obtenidos los ángulos alfa y gamma, se calcula beta de acuerdo al procedimiento explicado en el capítulo 5, sección 5.3.1. Con base en el valor del ángulo beta el motor de dirección girará un número determinado de pasos hacia el sentido establecido para cada caso. Debido a que el motor de dirección proporciona un movimiento de  $7.5^\circ$  por paso, y para conseguir una desviación significativa del móvil, se establecieron los rangos de movimiento respecto al valor del ángulo beta mostrados en la tabla 6.2.

| Beta                      | Entero | Estados | Ángulo de giro |
|---------------------------|--------|---------|----------------|
| [ $0^\circ - 13^\circ$ )  | 0      | 1       | $0^\circ$      |
| [ $13^\circ - 26^\circ$ ) | 1      | 3       | $15^\circ$     |
| [ $26^\circ - 39^\circ$ ) | 2      | 5       | $30^\circ$     |
| [ $39^\circ - 52^\circ$ ) | 3      | 7       | $45^\circ$     |
| [ $52^\circ - 65^\circ$ ) | 4      | 9       | $60^\circ$     |
| [ $65^\circ - 78^\circ$ ) | 5      | 11      | $75^\circ$     |
| [ $78^\circ - 90^\circ$ ) | 6      | 13      | $90^\circ$     |

Tabla 6.2 Ángulos de Giro del Motor de Dirección Respecto al Valor de Beta

El procedimiento para obtener el ángulo de giro del motor de dirección a partir de los intervalos definidos para beta es dividir el valor del ángulo beta entre trece (tamaño del intervalo), conservando el entero. Posteriormente, se duplica este entero y se incrementa en uno, lo que resulta en el número de estados necesarios para obtener el giro correspondiente. El valor de estados lo utiliza el programa como un contador que se decrementa cada vez que se escribe una combinación en el puerto B del microprocesador para mover el motor de dirección hasta que su valor es cero. Se establece un tiempo de espera para que el móvil corrija su trayectoria y que el motor de dirección regrese a su posición inicial.

Este tiempo depende de la velocidad que presenta el móvil, ya que entre mayor sea ésta, menor será el tiempo que requiera para corregir su trayectoria. La velocidad del móvil no se modificará durante la corrección de trayectoria.

Para comprobar que los ángulos calculados por el microprocesador fueran correctos y correspondieran a los valores registrados en la tabla 6.2, se realizó una prueba en la que diferentes coordenadas de destino fueron introducidas al sistema mientras el receptor GPS permaneció estático. De este modo el microprocesador utilizó sólo el ángulo alfa para el cálculo del ángulo beta pues el valor del ángulo gamma sería siempre cero. Las coordenadas del destino que se introdujeron al sistema están relacionadas con valores predeterminados de ángulos que el microprocesador debió calcular para poder verificar su valor. Se le llama a esta prueba la prueba de constelación por su apariencia. A continuación se presenta la figura 6.2 con una de las pruebas de constelación realizadas. Enseguida, se describe la manera en que debe interpretarse esta gráfica.

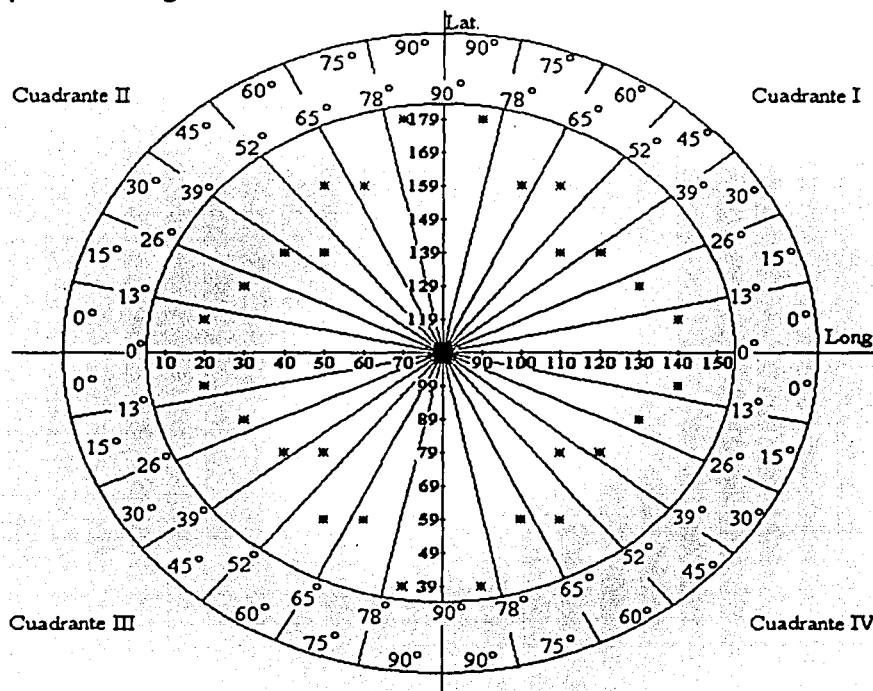


Figura 6.2 Gráfica de la Prueba de Constelación

En el centro de la gráfica aparecen las coordenadas de la posición del receptor GPS: sobre el eje horizontal la longitud y sobre el eje vertical la latitud. La posición del receptor es entonces  $80^\circ$  de longitud y  $109^\circ$  de latitud referidos al sistema absoluto de posicionamiento. Sobre este centro se trazan dos circunferencias; la circunferencia interior presenta varias divisiones que corresponden a los intervalos en que se distribuyó el rango de valores permitidos del ángulo beta. Cada cuadrante tiene siete divisiones de  $13^\circ$  cada una, comenzando por  $0^\circ$  hasta  $90^\circ$ . Los ángulos se miden desde el semieje horizontal más cercano y las diferencias de latitud y longitud entre la coordenada final de destino y la posición del receptor indican el sentido de giro o signo del ángulo en cada caso.

La circunferencia exterior en la figura 6.2 presenta otras divisiones, éstas corresponden a los intervalos del ángulo de giro en que se convierte el ángulo beta tal como se presentó en la tabla 6.2. Por ejemplo, todo ángulo marcado en la circunferencia interior cuyo valor se encuentre entre  $13^\circ$  y  $26^\circ$ , producirá un ángulo de giro de  $15^\circ$ . Si el ángulo interior se encuentra entre  $65^\circ$  y  $78^\circ$ , producirá un ángulo de giro de  $75^\circ$  en el motor de dirección.

Puesto que el valor del ángulo gamma es cero en todos los casos (ya que al permanecer estático el receptor GPS no hay diferencia de latitud o de longitud entre su posición inicial y su posición instantánea), cuando el microprocesador calcula el valor del ángulo beta en realidad está calculando el valor del ángulo alfa, que corresponde al ángulo medido entre la posición instantánea del receptor y la posición de destino o final. En la interfaz de comunicación del programa IASM11 (sección 6.2) se pueden monitorar el valor del ángulo alfa, los registros de control para el cálculo del ángulo beta, el número de pasos del motor de dirección y el ángulo de giro calculado por el microprocesador.

Introduciendo distintas coordenadas de destino (representadas por asteriscos en la gráfica de constelación) para las cuales se conocía previamente el valor del ángulo alfa que les corresponde, se pudo verificar que los cálculos del microprocesador fueran correctos. Por medio de esta prueba se comprobó que el ángulo alfa estuviera bien calculado y que los distintos casos posibles para el cálculo del ángulo beta fueran reconocidos adecuadamente por el programa en el microprocesador. Para verificar que el cálculo del ángulo gamma fuera igualmente correcto, se realizó una prueba distinta que se presenta en el capítulo 7 de este escrito.

Durante el tiempo dedicado a la corrección de trayectoria, el programa deshabilita automáticamente los sensores mediante el interruptor C1 para evitar cualquier interrupción ocasionada por estos. Se toma esta medida con el fin de prevenir una alteración de la trayectoria del móvil desde una posición del motor de dirección distinta de su posición inicial, ya que esto ocasionaría la pérdida del seguimiento de la trayectoria, o bien, un valor del ángulo de giro mayor que el límite permitido de  $90^\circ$ . Una vez corregida la trayectoria, el microprocesador habilita nuevamente el sistema de sensores permitiendo que ocurran interrupciones generadas por este sistema; así el programa vuelve a tomar otra lectura del receptor GPS para entrar en un ciclo de corrección de trayectoria hasta el momento en que se cumpla la condición límite de acercamiento al punto de destino.

Debido a la precisión del GPS (ver capítulo 7 sobre pruebas al respecto) y a las características del móvil, se decidió utilizar un movimiento de tipo celular para el recorrido de la trayectoria desde el punto inicial hasta el punto final, esto significa que el rastreo de la posición del móvil (toma de lecturas del receptor GPS) se realiza por segmentos o células sobre la trayectoria recorrida, con el fin de evitar errores en el cálculo de ángulos de corrección de la misma debidas a posicionamientos inexactos del móvil. La figura 6.3 ilustra la idea.

La trayectoria que el móvil describe de acuerdo al GPS se deriva del radio o zona de precisión del mismo. Usando el GPS, un punto en el espacio se verá representado por una circunferencia de 4 (m) de radio pues sabemos que la localización de un punto cualquiera tiene un error de precisión asociado de aproximadamente 4 (m) en cualquier dirección. De este modo se traza una línea uniendo estos 'puntos GPS' y así se obtiene la trayectoria del móvil. Sin considerar las dimensiones del móvil, puede trazarse otra trayectoria uniendo puntos que representan la posición real del móvil. La diferencia entre la posición real del móvil y la reportada por el GPS es el error de precisión que afectó el cálculo de la posición GPS en cada lectura particular. Por ejemplo, en el punto 1 de la figura 6.3 el error fue mínimo, pues la posición real se encuentra casi en el centro del área de precisión del GPS. En el punto 5 el error fue mayor, pues el punto que marca la posición real se encuentra muy alejado del centro de la circunferencia.

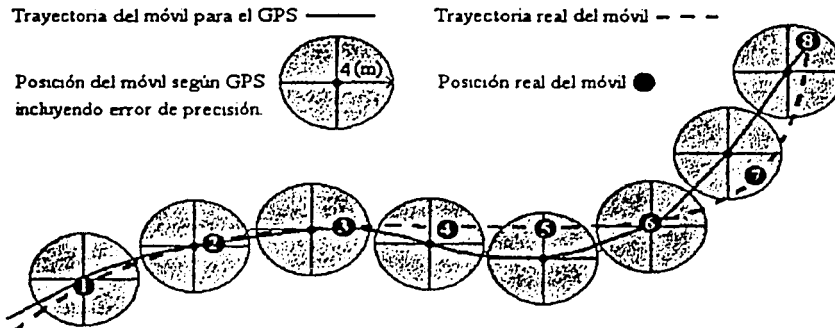


Figura 6.3 Rastreo de la Posición del Móvil en Forma Celular

Se observa que pueden existir diferencias entre las trayectorias sin que ello implique un problema mayor en lo que al sistema de navegación respecta. Esta condición se cumple siempre que los 'puntos GPS' o radios de precisión GPS no se confundan entre sí, es decir, que las circunferencias que definen la zona de precisión del GPS no se traslapen. De lo contrario, se tendrán errores mayores en el trazo de la trayectoria como se aprecia en la figura 6.4.

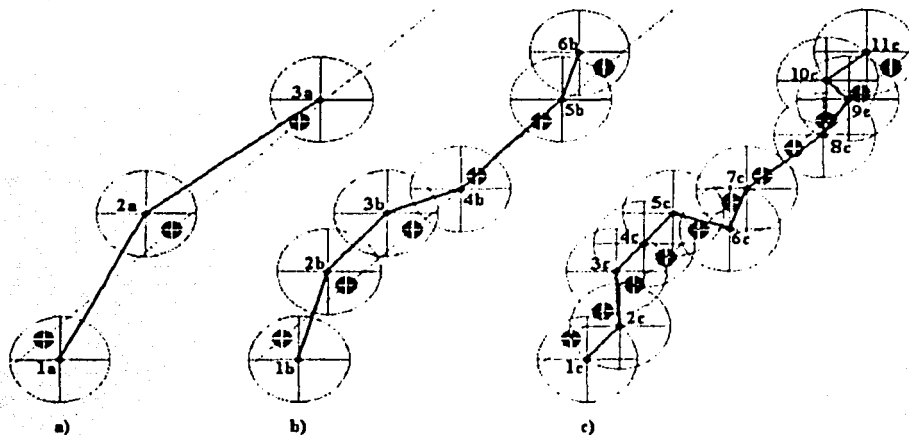


Figura 6.4 Diferentes Trayectorias GPS con Base en el Tiempo de Muestreo:

a) Tiempo de muestreo =  $T$ ; b) Tiempo de muestreo =  $T/2$ ; c) Tiempo de muestreo =  $T/4$

En los tres casos mostrados en la figura 6.4, a) b) y c), el recorrido real del móvil fue una trayectoria recta en un ángulo de  $45^\circ$  respecto a la horizontal. En la trayectoria a) el tiempo de muestreo fue lo bastante grande como para garantizar que en ninguna lectura la zona de precisión GPS se traslapara. La trayectoria GPS correspondiente se dibujó con una línea gruesa uniendo los puntos 1a, 2a y 3a que indican la posición del móvil reportada por el GPS. En la trayectoria b) se redujo el tiempo de muestreo a la mitad, por ello se tienen el doble de puntos GPS (1b, 2b, 3b, etc.). En este caso la zona de precisión GPS se traslapó en la mayoría de las lecturas con un porcentaje variable en cada ocasión. Para la trayectoria c) se redujo nuevamente el tiempo de muestreo a la mitad de modo que la zona de precisión GPS se traslapó en todas las lecturas considerablemente.

Lo que es importante observar de estos ejemplos es el grado de desviación que cada trayectoria presenta respecto a la que se considera que es la trayectoria real. Esto se registra en la tabla 6.3.

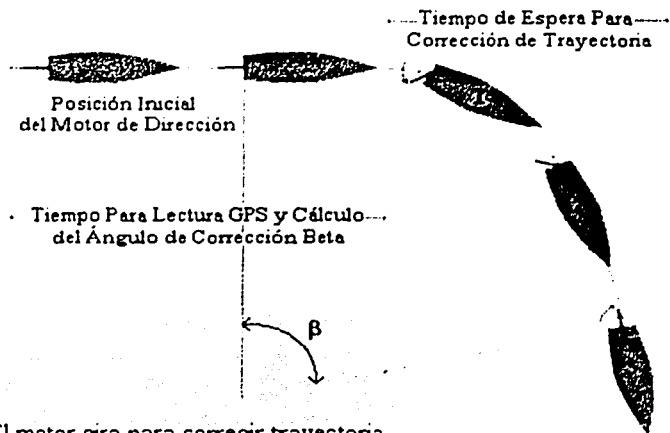
| a) Tiempo de muestreo = t |            | b) Tiempo de muestreo = t/2 |            | c) Tiempo de muestreo = t/4 |            |
|---------------------------|------------|-----------------------------|------------|-----------------------------|------------|
| Ángulo                    | Desviación | Ángulo                      | Desviación | Ángulo                      | Desviación |
| 63°                       | 18°        | 73.5°                       | 28.5°      | 49.8°                       | 4.8°       |
| 37.5°                     | 7.5°       | 49.6°                       | 4.6°       | 93.5°                       | 49.5°      |
|                           |            | 21.3°                       | 23.7°      | 47.3°                       | 2.3°       |
|                           |            | 46°                         | 1°         | 50.4°                       | 5.4°       |
|                           |            | 70°                         | 25°        | -19.6°                      | 64.6°      |
|                           |            |                             |            | 67.9°                       | 22.9°      |
|                           |            |                             |            | 38.9°                       | 6.1°       |
|                           |            |                             |            | 59.5°                       | 14.5°      |
|                           |            |                             |            | 135°                        | 90°        |
|                           |            |                             |            | 41°                         | 4°         |

Tabla 6.3 Desviación Entre Distintas Trayectorias GPS Comparadas con la Trayectoria Real.

En la tabla 6.3 se muestran los ángulos formados por cada segmento de cada trayectoria GPS respecto a la horizontal. La desviación se da en valor absoluto respecto a los 45°. Puede verse que la mayor desviación se presenta en la trayectoria c) como se esperaba, llegando incluso a los 90°. Además, encontramos también que uno de los ángulos medidos para la trayectoria c) resultó negativo, es decir que se midió debajo de la horizontal de referencia. La conclusión es que para esta trayectoria no sólo se presentan errores de magnitud en los ángulos sino también de lógica (por el sentido de los ángulos registrados) en la trayectoria.

Puede demostrarse fácilmente que incrementando el tiempo de muestreo en la señal GPS (es decir, tomando menos muestras) se disminuye el error en el cálculo de los ángulos de trayectoria del móvil y por consiguiente el error en el cálculo de los ángulos de corrección de la misma. Sin embargo, no debe caerse en el exceso pues se debe recordar que en su movimiento el móvil se encuentra sujeto a muchos elementos que perturban su trayectoria, por lo que un rastreo efectivo de la misma obliga a que la tasa de muestreo no sea demasiado baja (que el tiempo no sea muy alto) o de lo contrario el control del móvil será poco eficiente. Entre cada lectura GPS que se utiliza para el rastreo y corrección de la trayectoria del móvil debe existir un tiempo  $t$  determinado que garantice que la zona de precisión GPS no se traslape entre lecturas.

Con la intención de prevenir que el móvil pase de largo el punto de destino al no tomar una lectura oportuna del receptor GPS, se toman cinco lecturas intermedias entre las lecturas que producen una corrección de trayectoria del móvil y se verifica si se ha alcanzado el límite de radio de aproximación. Este número de lecturas puede ajustarse para que responda a las características de velocidad del móvil. La figura 6.5 representa la ejecución de una corrección en la trayectoria del móvil. A continuación, en la figura 6.6 se muestra el diagrama de flujo del programa principal en donde se observa el ciclo descrito. Cada proceso representado corresponde a una subrutina del programa principal, excepto los que toman lugar cuando el móvil ha alcanzado su destino, que son parte de una misma subrutina y por ello se representan englobados en un solo cuadro externo de proceso. Además, el pequeño ciclo donde se decrementa  $N$  y posteriormente se compara con cero, pertenece directamente al programa principal, es decir, no corresponde a una subrutina.



1. El motor gira para corregir trayectoria.
2. Se conserva la posición del motor mientras se completa la corrección.
3. El motor regresa a su posición inicial, terminó corrección de trayectoria

Figura 6.5 Proceso de Corrección de Trayectoria del Móvil

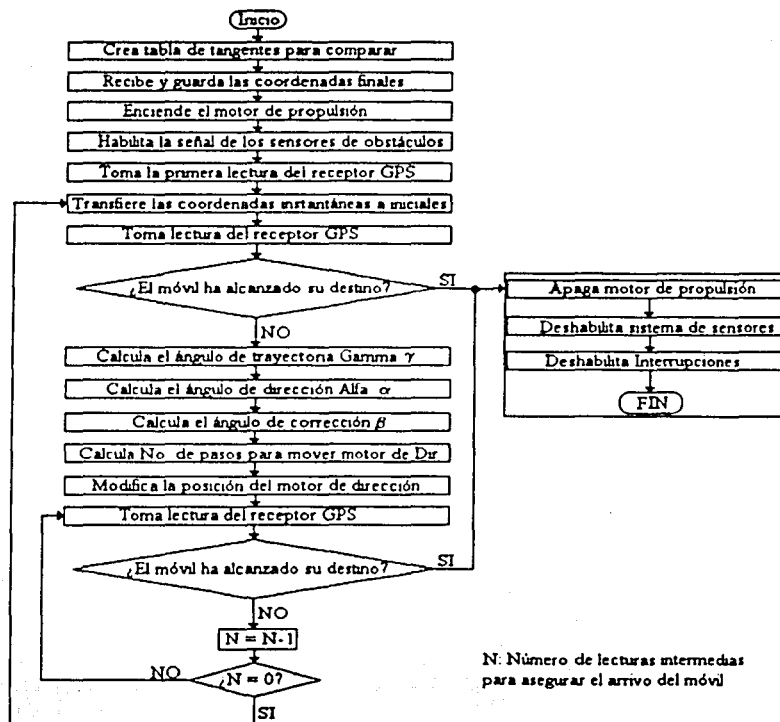


Figura 6.6 Diagrama de Flujo del Programa Principal del Sistema

### 6.3.3 Interrupción Por Sensores

El programa principal se encuentra sujeto a dos tipos de interrupciones; la primera es interrupción por software (SWI) que genera la señal de ciclo de trabajo definido para controlar el motor de propulsión, y se encuentra habilitada durante toda la ejecución del programa, excepto cuando ocurre el otro tipo de interrupción, que es una interrupción externa enmascarable activada por el *pin* IRQ conectado al sistema de sensores (subsección 4.1.3).

Cuando el sistema de sensores detecta un obstáculo, se genera una señal que se envía al microprocesador de modo que ocurre una interrupción externa. Dependiendo de la combinación de sensores que se hayan activado, la trayectoria del móvil se modificará de acuerdo con la tabla 6.4 en donde un sensor activado se representa por 1 y un sensor desactivado por 0.

| Caso | Sensor Izq. | Sensor Central | Sensor Der. | Respuesta del móvil.  |
|------|-------------|----------------|-------------|---|
| 1    | 1           | 0              | 0           | Disminuye vel. y espera a que se presente otra combinación. |
| 2    | 0           | 0              | 1           | Disminuye vel. y espera a que se presente otra combinación. |
| 3    | 0           | 1              | 0           | Disminuye vel. y gira 15° a la Izquierda.                   |
| 4    | 1           | 1              | 0           | Disminuye vel. y gira 60° a la Derecha                      |
| 5    | 0           | 1              | 1           | Disminuye vel. y gira 60° a la Izquierda                    |
| 6    | 1           | 1              | 1           | Disminuye vel. y gira 90° a la Derecha                      |

**Tabla 6.4 Respuesta del móvil ante una interrupción de los sensores de obstáculos**

En las primeras dos combinaciones de la tabla 6.4, en las que sólo se activa un sensor, el programa termina la interrupción y regresa al programa principal esperando que una nueva interrupción se genere por una combinación de al menos dos sensores activados. Esto se hace con la intención de asegurar la presencia de un obstáculo en la trayectoria del móvil y no modificar esta última por un objeto cercano que no obstruye el camino del móvil. Debe quedar claro que el sistema de sensores únicamente indica la presencia de un objeto dentro del rango de detección de los sensores ultrasónicos, no arroja ninguna información respecto a las dimensiones del objeto ni ninguna otra característica propia del mismo.

La respuesta del móvil en el resto de las combinaciones de sensores en la tabla 6.4 se diseñó considerando el ángulo de detección de los sensores (60°), de modo que con el movimiento, aquel sensor que se haya activado alcance a librar el obstáculo detectado y el móvil quede libre en su trayectoria.

Al igual que en el caso de la corrección de trayectoria para navegación, se incluye un tiempo de espera para que el móvil pueda evadir un obstáculo y que el motor de dirección regrese a su posición inicial. Durante este tiempo de espera se deshabilita el sistema de sensores y el ciclo de trabajo del motor de propulsión se reduce al 50 % pues la interrupción por software que lo genera también se deshabilita.

Es importante mencionar que el sistema no fue diseñado para llevar un rastreo de la posición del móvil durante una interrupción de los sensores. La razón es que no es posible prever el momento en el que el programa principal será interrumpido por una lectura de los sensores, de modo que no se conoce el proceso que será interrumpido y por tanto durante la interrupción no deben llevarse a cabo operaciones que modifiquen el flujo de esos procesos como sería la toma de una lectura del sensor GPS.

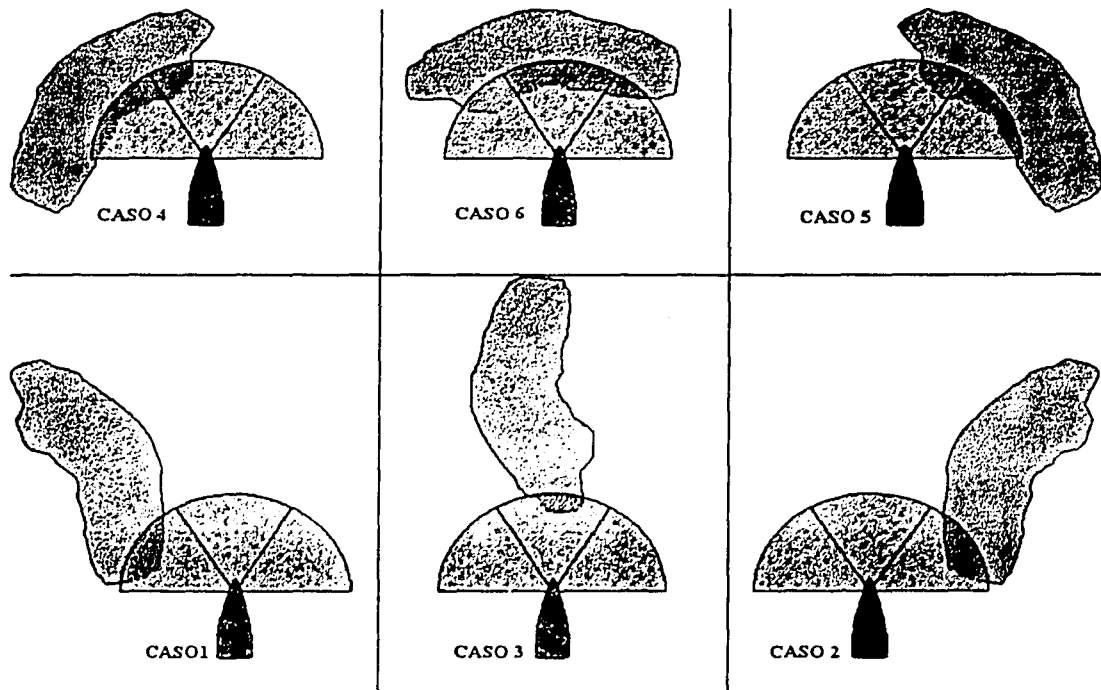


Figura 6.7 Combinaciones Para el Movimiento de Evasión de Obstáculos

Luego de una interrupción por sensores, el programa continúa con el proceso que estaba ejecutando antes de ser interrumpido. Puede ser que el programa estuviera calculando el ángulo de dirección  $\alpha$  en el momento que fue interrumpido, de modo que al salir de la interrupción terminará el cálculo y procederá con otros cálculos. Se observará que los cálculos derivados de esa situación serán incorrectos, pues el móvil, luego de una interrupción, ya no se encuentra en la posición tomada para realizar los cálculos. Este problema, sin embargo, no afecta gravemente el desempeño del sistema pues bastará esperar poco tiempo para que se tome una nueva lectura GPS y se calculen nuevamente las correcciones pertinentes en la trayectoria del móvil.

#### 6.4 IMPLEMENTACIÓN DE PROTOTIPOS

Para concluir este capítulo, se presentan imágenes de los dispositivos utilizados con el fin de ilustrar sus dimensiones, las conexiones realizadas entre subsistemas y las condiciones en las cuales se desarrollaron el diseño y la depuración del sistema. También se presenta un diagrama general de entradas y salidas al microprocesador donde aparecen todos los elementos del sistema de navegación diseñado.



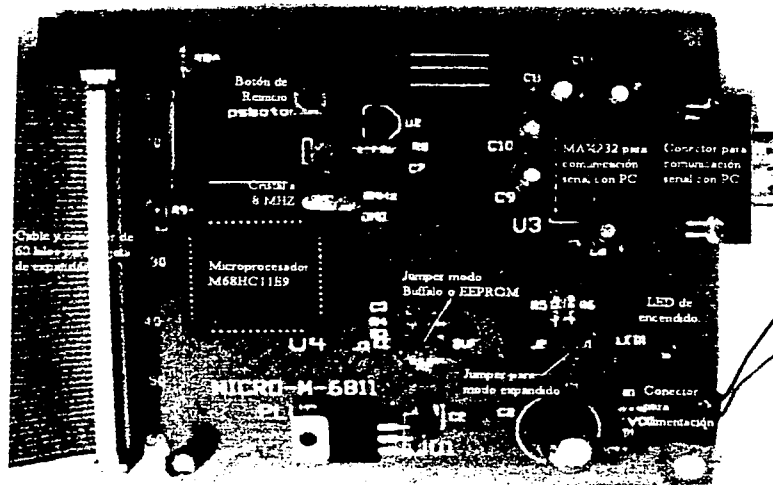


Figura 6.7 Tarjeta de Desarrollo del Microprocesador M68HC11.  
Dimensiones: 10.5 (cm) x 12.5 (cm).

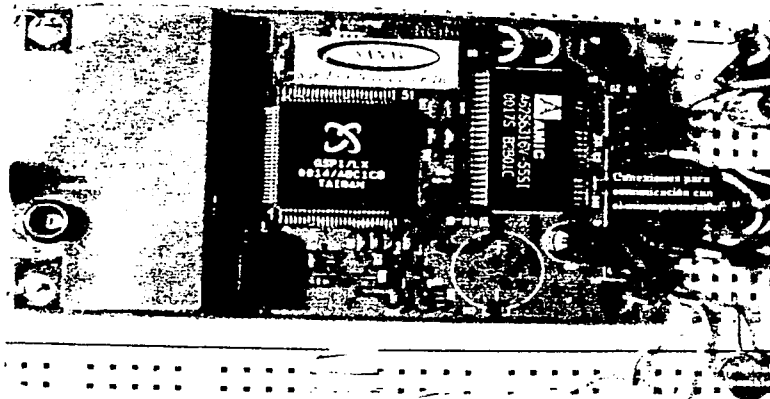


Figura 6.8 Tarjeta del Receptor GPS FV-12  
Dimensiones: 7.1 (cm) x 4.1 (cm)

TESIS CON  
FALLA DE ORIGEN

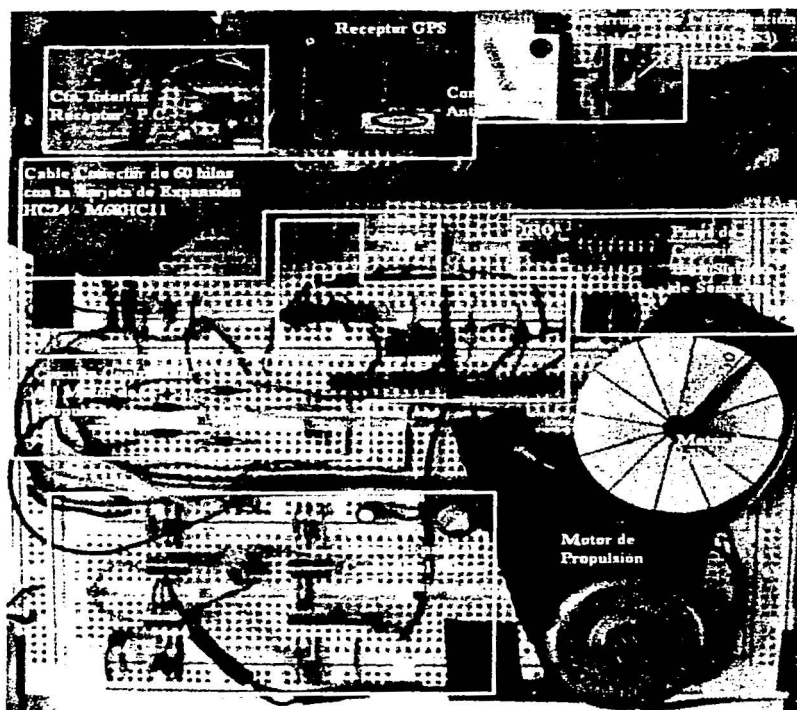


Figura 6.9 Implementación de los Circuitos de Control de Hardware Prototipo

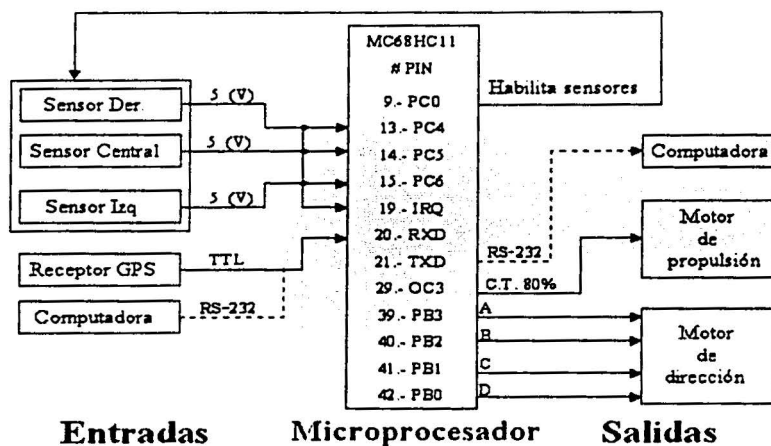


Figura 6.10 Diagrama General de Entradas y Salidas del Sistema

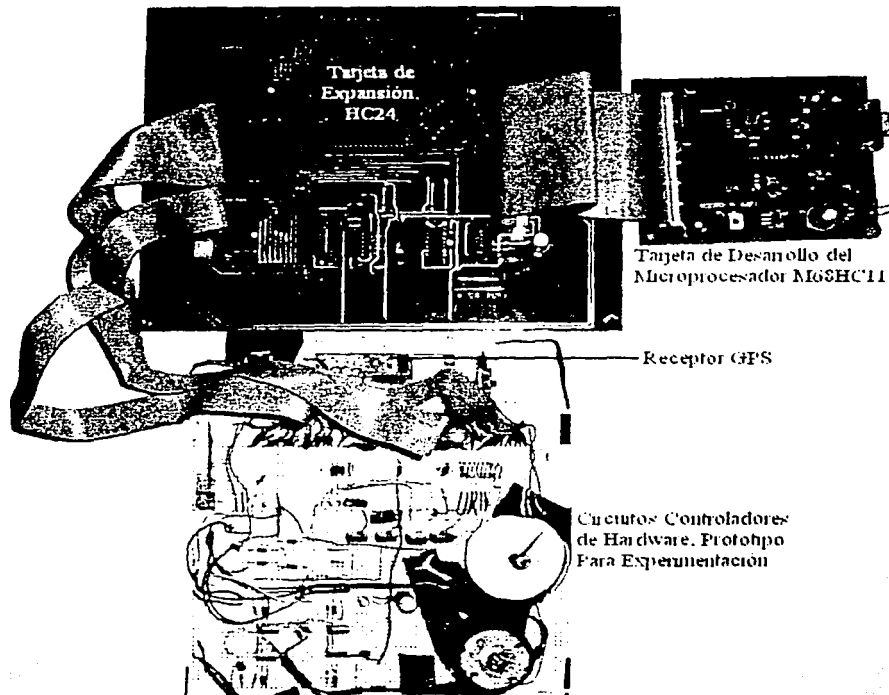


Figura 6.11 Prototipo del Sistema de Navegación

### REFERENCIAS BIBLIOGRÁFICAS

- [1] Doughman, Gordon; "MC68HC11 Floating Point Package"; 1986;  
<http://www.nixu.fi/~lauri/hc11loki/fp11/hc11fp.doc.html>

## **PRUEBAS Y RESULTADOS**

Durante el desarrollo del proyecto se realizaron diversas pruebas a cada uno de los subsistemas de manera independiente. Estas pruebas se llevaron a cabo como parte del diseño del sistema, con la finalidad de evaluar el desempeño de cada elemento en particular y para garantizar una buena integración de los subsistemas.

En una etapa posterior a la integración del sistema completo, se practicaron las pruebas dirigidas a la evaluación de su desempeño. Como se adelantó en la introducción de este escrito, estas pruebas se agrupan en tres tipos: pruebas de monitoreo del sistema, pruebas de desempeño bajo condiciones controladas y pruebas de desempeño autónomo. En este capítulo se presentan estas pruebas con los resultados y conclusiones que de éstas se desprenden.

### **7.1 PRUEBAS DE MONITOREO DEL SISTEMA**

#### **7.1.1 Pruebas De Rastreo De Trayectorias**

Las primeras pruebas practicadas sobre el sistema en su etapa de depuración se hicieron con la intención de verificar que el funcionamiento del receptor GPS fuera correcto en el rastreo de trayectorias, ya que hasta esta etapa sólo se le había probado en condiciones estáticas, es decir, sin desplazamiento del mismo. Se recopilaron lecturas de las coordenadas recibidas por el receptor GPS mientras éste se desplazaba en varias direcciones a bordo de un automóvil; posteriormente se trazaron mapas de las coordenadas recopiladas y se compararon con otros mapas del área trazados por otros medios.

Se recorrieron segmentos de los circuitos existentes dentro de Ciudad Universitaria, pues se dispuso de planos del área con la suficiente resolución y precisión como para usarlos en la comparación con las lecturas del receptor GPS graficadas. Se buscó mantener la velocidad del automóvil en un promedio de 20 Km/Hr (5.5 m/s), sin embargo, esto no fue posible en todos los tramos del recorrido debido al tráfico presente en los circuitos. Las lecturas del receptor GPS se tomaron aproximadamente cada 6 segundos.

Se recopilaron lecturas de coordenadas por medio del GPS en varias ocasiones, en diferente hora del día y bajo distintas condiciones climáticas (lluvia ligera, zonas de alta concentración de vegetación, y con alta temperatura por luz solar) observando que el funcionamiento del receptor GPS no se vio afectado de forma apreciablemente. Aunque las condiciones en que se probó el receptor no fueron las más adversas posibles, puede concluirse que el GPS funciona bien en un amplio rango de posibilidades climáticas. Las figuras 7.1 y 7.2 presentan dos de las pruebas de rastreo de trayectorias, con el fin de ilustrar los resultados que se obtuvieron al respecto.



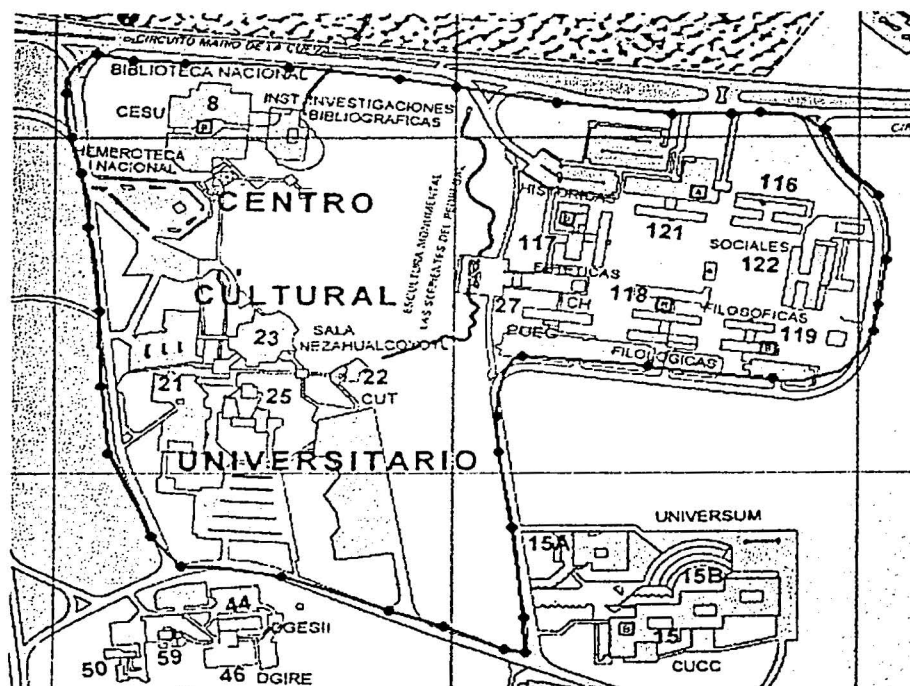


Figura 7.2 Prueba de Rastreo de Trayectoria, Centro Cultural Universitario

En la figura 7.2 se presenta un recorrido realizado en la zona del Centro Cultural Universitario. En comparación con la figura 7.1 del recorrido sobre el circuito escolar, la figura 7.2 se encuentra en una escala mayor, lo que significa que el recorrido en este caso fue de menores dimensiones. Los errores de precisión de las lecturas tomadas del receptor GPS son más notorios en este caso. Por ejemplo, pueden destacarse los puntos que se encuentran en la parte derecha del recorrido, que aparecen como si el receptor se hubiese desplazado sobre los edificios de Investigaciones Filológicas, lo que por supuesto nunca ocurrió. Asimismo, muchos de los puntos salen ligeramente del circuito, ya sea hacia el exterior o hacia el interior del mismo.

Estas discrepancias deben atribuirse principalmente a tres fuentes: el error en la precisión del receptor GPS; el error en el ajuste de la escala del mapeo de los puntos GPS respecto a la escala del plano usado como comparación; y el error en el trazo del plano de comparación.

Pese a estas pequeñas diferencias puede concluirse que el receptor GPS es adecuado para utilizarse en sistemas de navegación o localización de mediana precisión. Para comprobarlo y para complementar las pruebas de rastreo de trayectorias, se llevaron a cabo pruebas de caracterización del receptor GPS, con el fin de medir su precisión experimentalmente.

### 7.1.2 Pruebas De Precisión Estática

Para medir la precisión real del receptor GPS se mantuvo al receptor fijo en un punto y se recopilaron las lecturas recibidas por el mismo. Las coordenadas reportadas por el receptor mostraron una pequeña variación conforme pasó el tiempo, y al cabo de un número suficiente de lecturas (alrededor de 100 o 150) del mismo punto, se desactivó el receptor y se analizaron las lecturas obtenidas. Las gráficas que se presentan en las figuras 7.3 y 7.4 se realizaron con base en esas lecturas. La prueba 1 consta de 100 muestras, y se presentan por separado los valores de longitud (prueba 1a, figura 7.3) y de latitud (prueba 1b, figura 7.4) de un mismo punto geográfico. El valor más cercano al centro del círculo en la gráfica es el menor, y el más alejado el mayor.

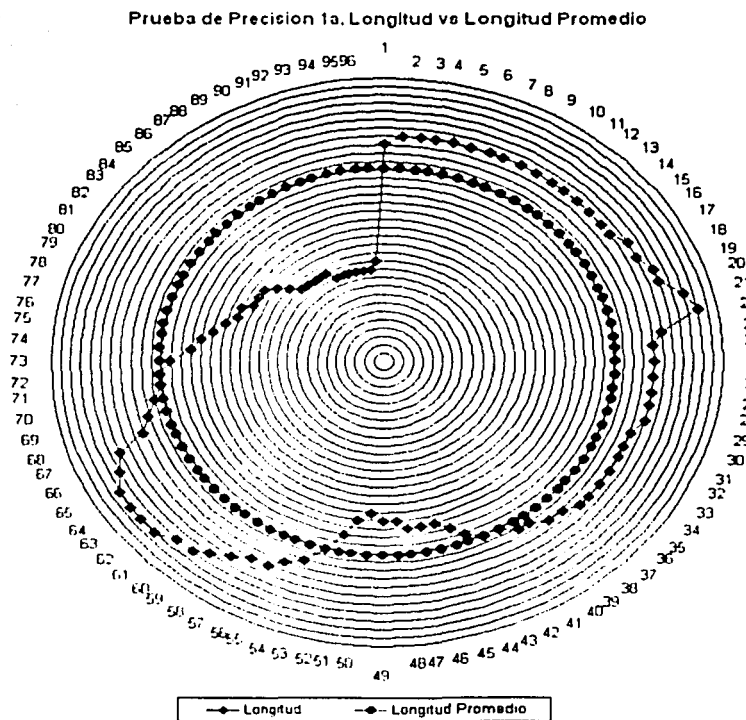
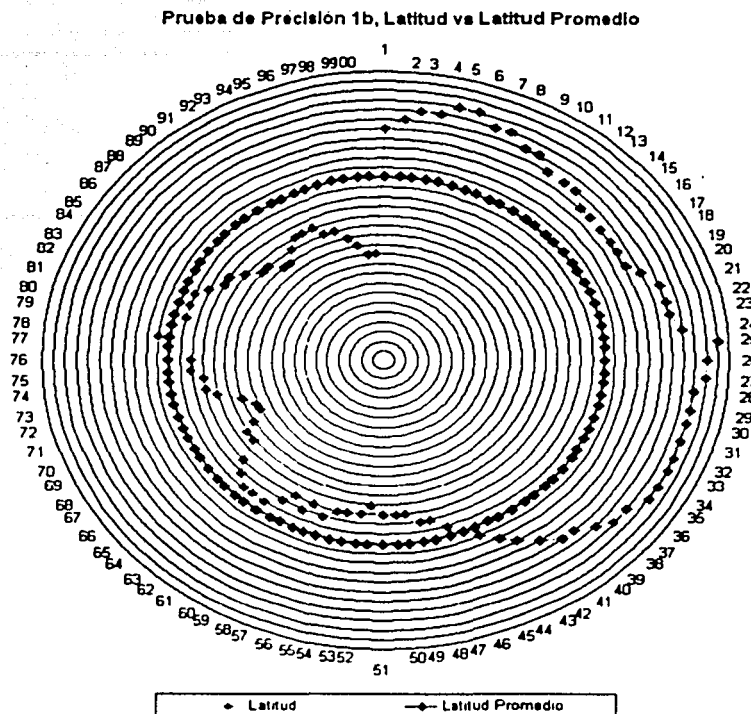


Figura 7.3 Prueba 1a de Precisión Estática



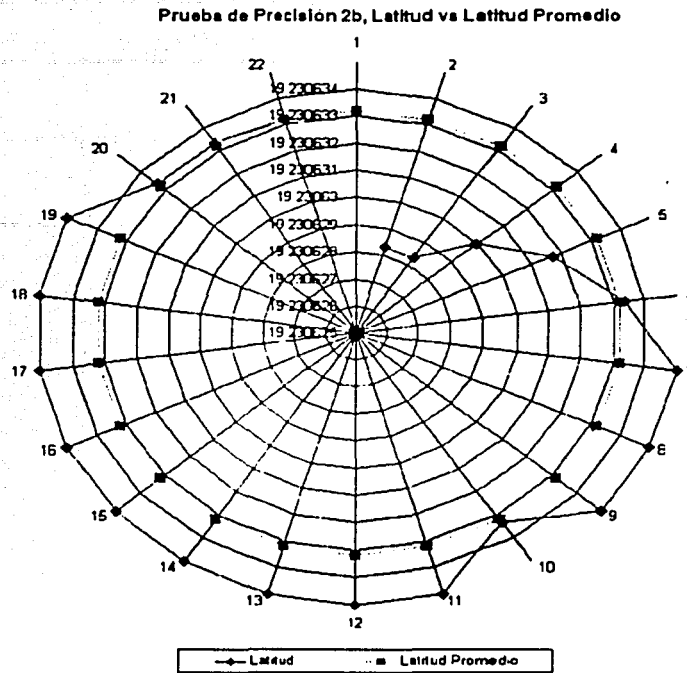
**Figura 7.4 Prueba 1b de Precisión Estática**

De las pruebas de precisión estática del receptor, se observa que desde el punto de vista estadístico el receptor presenta una muy buena precisión, que converge al valor de  $3.05E-05$  grados para la longitud, equivalente a 3.20 (m), y  $4.38E-05$  grados para la latitud, es decir 4.87 (m). La resultante de estas dos diferencias es de  $5.34079E-05$  grados, o 5.83 (m). Este resultado proviene de una prueba con 1000 muestras tomadas sobre el mismo punto con el receptor GPS.

Sin embargo, este valor de precisión no es el que debe considerarse para el diseño de un sistema de navegación puesto que para estos sistemas no se toman varias lecturas para luego utilizar un valor promedio, sino que se usa la lectura instantánea. Por ello se realizaron pruebas de precisión tomando en cuenta menos muestras, con la finalidad de evaluar la variación en los valores de coordenadas reportados por el receptor en un lapso de tiempo más corto aproximándonos a las condiciones dinámicas. Así, la prueba 2 consiste de 22 muestras. Las gráficas que se presentan en las figuras 7.5 y 7.6 reportan los valores correspondientes.







**Figura 7.6 Prueba 2b de Precisión Estática**

Comparando la gráfica de la figura 7.6 con la anterior, figura 7.5, se aprecia que la variación en los valores de latitud tiene una distribución diferente de la que se presentó para los valores de longitud. Lo mismo puede observarse en las gráficas de la prueba 1 de precisión, figuras 7.3 y 7.4, pero es más claro en el caso de la prueba 2.

Analizando las dos últimas gráficas, figuras 7.5 y 7.6, se logra llegar a una conclusión importante: los valores de las coordenadas, en longitud y en latitud, no varían abruptamente entre lecturas consecutivas. Esto significa que en una aplicación dinámica en la que el receptor tome lecturas de coordenadas con un intervalo de tiempo pequeño (entre 1 y 7 segundos), los valores de las lecturas consecutivas mantendrán coherencia entre sí, es decir, presentarán un error relativo similar.

Se confirma así que el receptor GPS puede utilizarse en aplicaciones dinámicas de mediana precisión (el error de precisión, aunque pequeño, no se elimina), y se puede añadir la condición de que la tasa de muestreo del receptor debe ser lo suficientemente alta, dependiendo de las características del sistema. Más adelante se verá que hay otra consideración importante derivada de las pruebas de precisión, que afectará el límite del radio de acercamiento definido para el sistema.

## **7.2 PRUEBAS DE DESEMPEÑO BAJO CONDICIONES CONTROLADAS**

Una vez evaluada la capacidad del sistema para el rastreo de trayectorias, se realizaron pruebas para confirmar que el sistema realizara adecuadamente las operaciones básicas para el cálculo de la corrección de trayectoria. En el capítulo 6 se presentó una prueba diseñada para verificar que el cálculo del ángulo alfa fuera correcto; enseguida se describen las pruebas realizadas para verificar que el cálculo del ángulo gamma también se aplicara correctamente.

### **7.2.1 Prueba De Identificación De Casos**

El ángulo de trayectoria gamma se calcula tomando las diferencias de longitud y latitud entre las coordenadas de la posición instantánea del móvil (S) y la posición inicial del mismo en un momento dado (I). Además del valor del ángulo es necesario conocer su sentido de modo que el sistema pueda reconocer la operación que ha de realizar para calcular el ángulo de corrección de trayectoria beta.

El ángulo gamma puede calcularse de acuerdo a cuatro casos generales que dependen de la posición relativa que ocupa S respecto a I (capítulo 5). Para asegurar que el ángulo gamma se calcula correctamente, debe comprobarse que el sistema reconoce adecuadamente cada uno de estos cuatro casos generales. De este modo, la prueba para verificar el cálculo del ángulo gamma consistió en hacer que el móvil recorriera una serie de trayectorias fácilmente identificables con la intención de verificar que el sistema reconociera cada caso correctamente. Esencialmente se utilizaron las trayectorias de prueba que se presentaron en la parte final del capítulo 5, incluyendo para cada caso del ángulo gamma (A, B, C y D) las cuatro posibilidades del ángulo alfa. Como resulta obvio, la prueba concluyó una vez que se confirmó que el sistema reconoce todos los casos posibles y sus combinaciones adecuadamente, pues el funcionamiento del sistema depende completamente de este supuesto. Se realizaron ajustes menores al programa de control y se obtuvieron dos conclusiones importantes:

- 1) Puesto que el programa de control se basa en los signos de las diferencias entre las posiciones del receptor GPS según sea el caso, es muy importante que la magnitud de estas diferencias sean lo suficientemente grandes para que no haya confusión en el sistema, ya que si el valor de la diferencia se aproxima mucho a cero por la izquierda (signo negativo), el valor será redondeado a cero (signo positivo) y se perderá el signo correspondiente. Se concluye que es necesario que las lecturas del sensor GPS que se utilizan para el cálculo de la corrección de trayectoria se encuentren separadas una distancia promedio de 10 o 12 (m) como mínimo con el fin de que la diferencia entre posiciones se aleje lo suficiente de cero, evitando así la confusión del sistema. El tiempo en la tasa de muestreo debe entonces ajustarse para conseguir esta condición según la velocidad propia del móvil.
- 2) Existe una excepción importante al punto anterior, que se presenta cuando el móvil se mueve con una trayectoria que coincide o se aproxima mucho a alguno de los dos ejes de referencia, latitud o longitud. En este caso alguna de las diferencias invariablemente será cero aun cuando la distancia entre lecturas del receptor GPS sea grande. El sistema tomará el caso que corresponde al signo positivo aunque el movimiento real haya sido el contrario. Claramente esta es una situación que causará un error en el cálculo de la corrección de trayectoria del móvil, aunque también es una situación que se presenta muy raramente y el error que produce es mínimo.

### 7.2.2 Prueba De Recorrido De Trayectoria Fija

Considerando que los resultados de las pruebas preliminares fueron positivos, se inició otro tipo de pruebas recorriendo una trayectoria conocida y verificando que las respuestas del sistema (la magnitud de los ángulos en cada caso) fueran las adecuadas en cada punto de lectura. Para ello se recogieron las coordenadas de varios puntos de la trayectoria, se calcularon los ángulos de corrección que el sistema debía obtener y se compararon con los ángulos de las pruebas. Un ejemplo aparece en la figura 7.7.

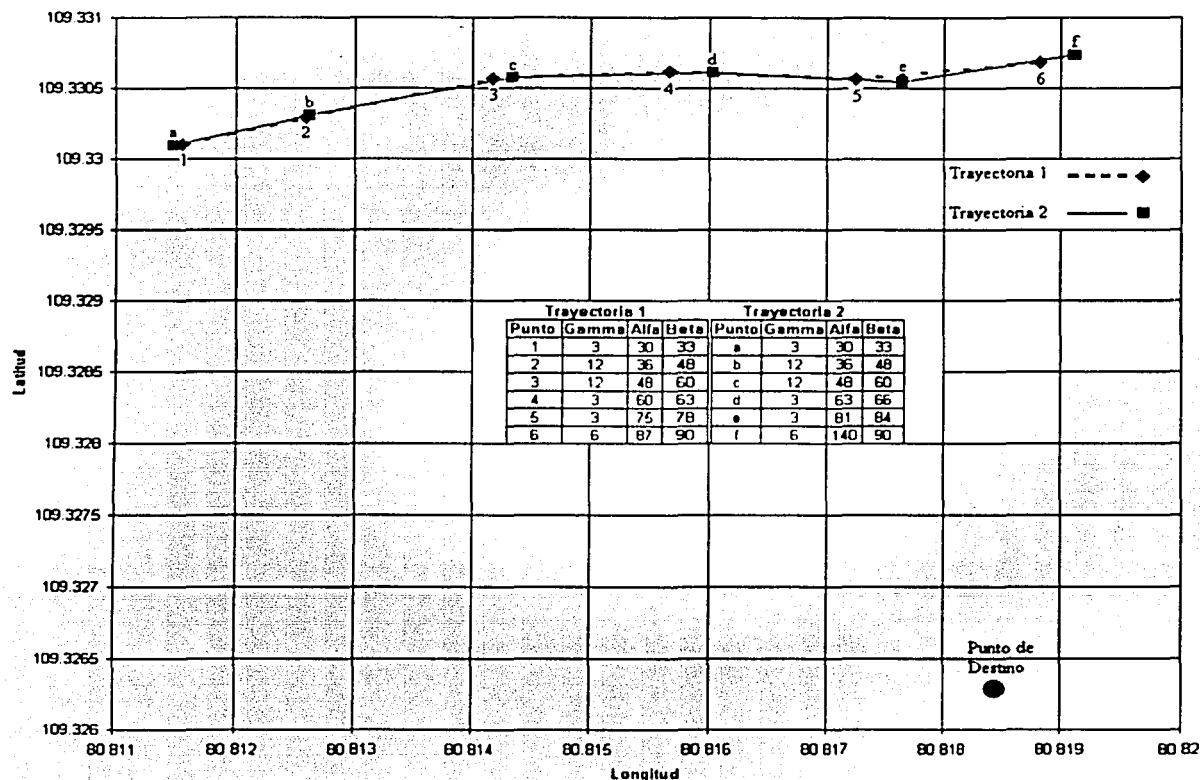


Figura 7.7 Prueba de Trayectoria Fija

En la figura 7.7 se muestran los valores de todos los ángulos que calculó el sistema; como estos valores son aproximados, en la tabla 7.1 se presentan los valores calculados con mayor precisión, que se utilizaron para comprobar los de la prueba.

| Trayectoria 1 |       |      |      | Trayectoria 2 |       |      |      |
|---------------|-------|------|------|---------------|-------|------|------|
| Punto         | Gamma | Alfa | Beta | Punto         | Gamma | Alfa | Beta |
| 1             | 0.0   | 29.0 | 29.0 | a             | 26.6  | 28.7 | 55.3 |
| 2             | 10.6  | 34.4 | 45.0 | b             | 10.5  | 34.6 | 45.0 |
| 3             | 9.6   | 45.0 | 54.6 | c             | 8.9   | 46.2 | 55.1 |
| 4             | 1.9   | 57.4 | 59.4 | d             | 1.2   | 60.7 | 62.0 |
| 5             | 1.8   | 74.3 | 76.1 | e             | 2.5   | 79.3 | 81.8 |
| 6             | 4.3   | 85.0 | 89.4 | f             | 7.5   | 81.4 | 88.9 |

Tabla 7.1 Valores Teóricos de Prueba de Trayectoria Conocida

La diferencia entre los valores de la tabla 7.1 y los de la figura 7.7 calculados por el sistema radica en que para los primeros la función 'ángulo cuya tangente' ( $\text{atan}(\theta)$ ) se aplicó directamente a los valores de las tangentes de los ángulos, mientras que para los valores prácticos el sistema usó una tabla de correspondencia construida con menor precisión (véase capítulo 6 sección 6.3.2). A pesar de esta diferencia, se comprobó que cuando los valores teóricos se comparan con los intervalos definidos en el sistema, se obtienen los valores prácticos reportados.

La conclusión que puede obtenerse con estas pruebas es que los ángulos de corrección que se calculen en el recorrido de cualquier trayectoria serán correctos, de modo que se podrán producir diferentes conclusiones acerca del comportamiento o desempeño autónomo del sistema con la seguridad de que el cálculo de los ángulos no afecta el mismo.

### 7.3 PRUEBAS DE DESEMPEÑO AUTÓNOMO

Tomando en cuenta que las pruebas realizadas para verificar el cálculo de los ángulos alfa, gamma y beta, así como la prueba del reconocimiento de casos fueron positivas, en esta sección se presentan las pruebas realizadas con la finalidad de evaluar el comportamiento autónomo del sistema en general. Se trata de dos tipos de pruebas: recorrido de trayectoria indicada por el sistema; y verificación del límite de acercamiento.

Estas pruebas se realizaron por separado debido a que la prueba de verificación del límite de acercamiento no permitía recabar los datos necesarios para la prueba de trayectoria. Además, para la verificación del límite de acercamiento no es necesario recorrer toda la trayectoria indicada por el sistema, pues basta recorrer una pequeña distancia desde una posición cualquiera, con dirección hacia el punto de destino definido y comprobar que el sistema reconoce el límite definido.

#### 7.3.1 Prueba De Recorrido De Trayectoria Indicada Por El Sistema

Para el desarrollo de esta prueba primeramente se seleccionó un área de terreno libre de obstáculos donde fuera posible efectuar los distintos desplazamientos que el sistema calcula para la corrección de una trayectoria dada. Dicha área debía ser lo suficientemente extensa como para permitir varias tomas de lectura del receptor GPS con una distancia entre lecturas de 12 (m) como mínimo para evitar confusiones en el reconocimiento de los distintos casos de orientación y disminuir los errores en el cálculo de los ángulos debidos a la precisión del GPS.

Se buscó que el terreno fuera lo más regular posible para facilitar el desplazamiento del vehículo y que su movimiento correspondiera efectivamente a los comandos generados por el microprocesador. Es así como se escogió un estacionamiento adyacente al Estadio Olímpico México 86 en Ciudad Universitaria. Dentro de este terreno se encontró un punto que sirvió como destino al móvil y a partir del cual se tuvo el mayor espacio disponible para diseñar distintas trayectorias que el sistema debió recorrer. Se registraron las coordenadas de ese punto y se definieron las trayectorias de prueba con base en un conjunto de movimientos que se consideran representativos para evaluar el desempeño autónomo del sistema.

Finalmente se colocó el móvil en el punto de origen de cada trayectoria definida y se inicializó el sistema de control de navegación; se introdujeron las coordenadas de destino y se comenzó el desplazamiento del móvil. Posteriormente, se observaron las correcciones de trayectoria calculadas por el sistema y se simuló su ejecución, modificando la dirección del móvil de acuerdo a esos valores. La prueba concluyó en el momento en que la diferencia calculada entre las posiciones final e instantánea del móvil fue menor que el límite de radio de acercamiento definido y el motor de propulsión se detuvo. Entonces, se registraron los valores de las posiciones que recorrió el móvil; los ángulos que calculó en su trayectoria; y los casos de orientación que identificó. A partir de estos parámetros, se trazaron gráficas de las trayectorias recorridas y se compararon los movimientos del móvil con aquellos que fueron planeados. Se recorrió cada trayectoria dos veces para confirmar los resultados del sistema. Las figuras 7.8 a 7.11 corresponden a las trayectorias de prueba registradas.

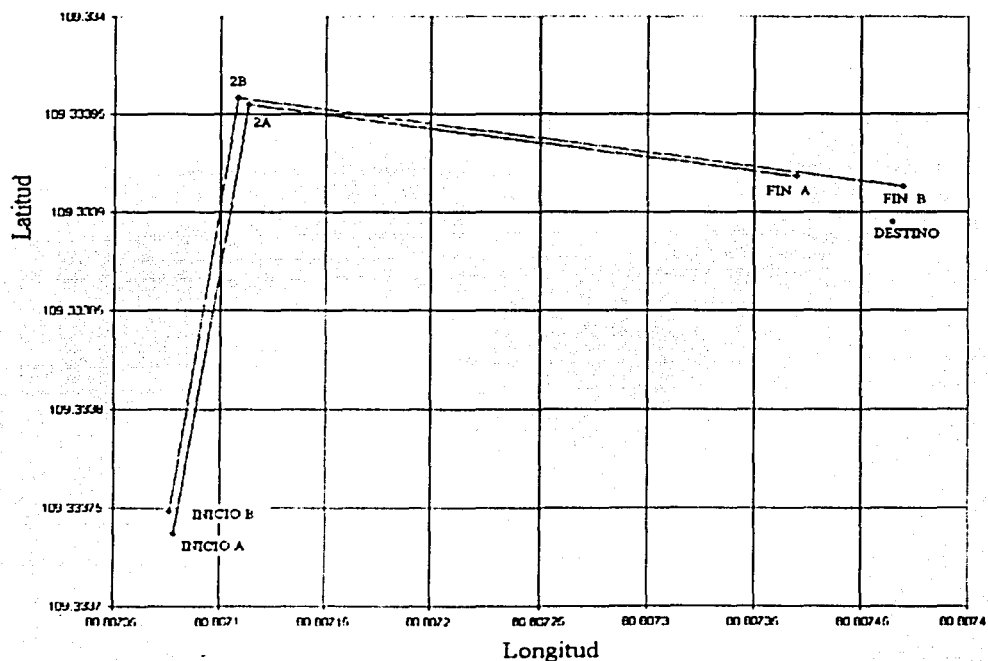


Figura 7.8 Trayectoria de Prueba 1

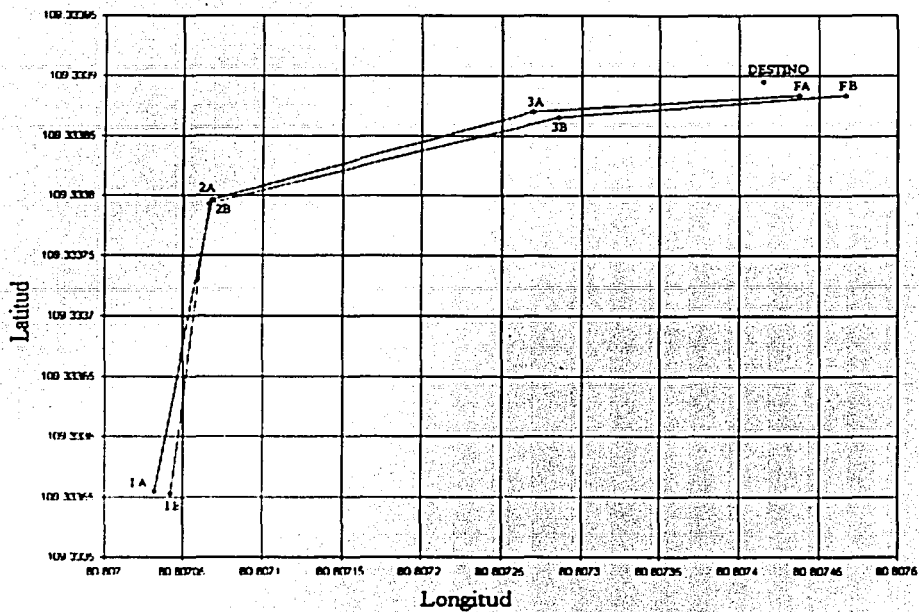


Figura 7.9 Trayectoria de Prueba 2

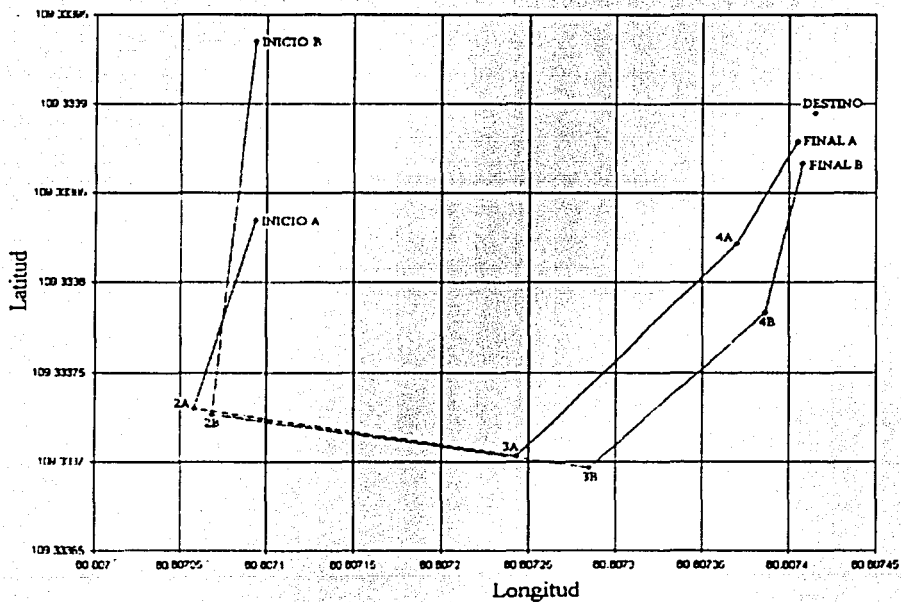


Figura 7.10 Trayectoria de Prueba 3

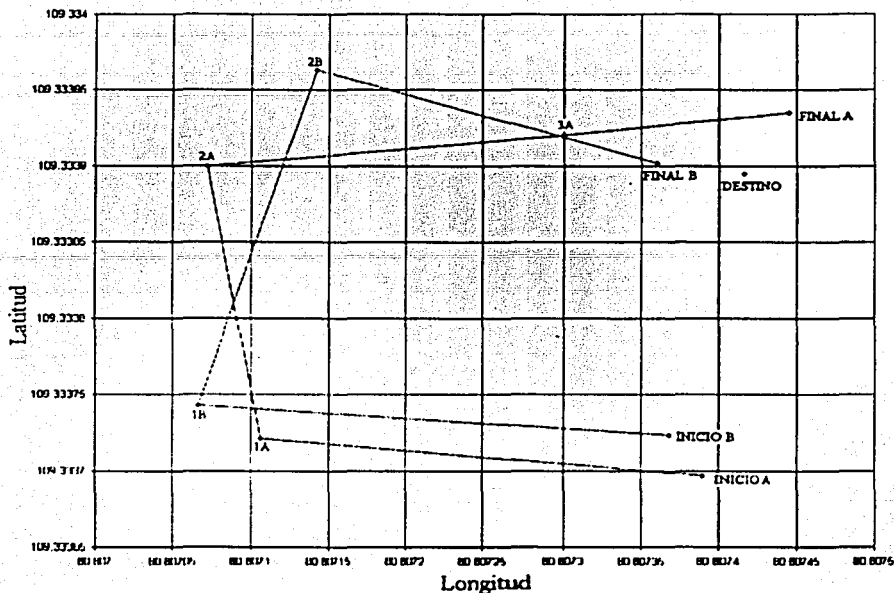


Figura 7.11 Trayectoria de Prueba 4

En las figuras de las pruebas 1 a 4, se observa que las trayectorias recorridas efectivamente guiaron al móvil hasta el punto de destino. En cada trayectoria se marcaron los puntos en que el sistema calculó una corrección de trayectoria. Se verificó que los casos de orientación y las magnitudes de los ángulos concordaran con aquellos que fueron planeados para cada trayectoria en particular. Adicionalmente, se realizó un análisis sobre cada conjunto de datos de las pruebas. Se muestra a continuación un ejemplo del análisis realizado sobre los datos recabados de las trayectorias de prueba, en particular se trata de los datos de la prueba 2. Estos datos aparecen en la tabla 7.2.

| Prueba 2, Recorrido A |                        |               |       |               |               |      |               |      |      |
|-----------------------|------------------------|---------------|-------|---------------|---------------|------|---------------|------|------|
| Punto                 | Coordenadas            | Dir. $\gamma$ | Gamma | Caso $\gamma$ | Dir. $\alpha$ | Alfa | Caso $\alpha$ | Beta | Giro |
| IA                    | 19G20.0133N99G11.5781W |               |       |               |               |      |               |      |      |
| 2A                    | 19G20.0278N99G11.5759W | 2B8Ch         | 84°   | A             | 2B19h         | 15°  | C             | 69°  | 75°  |
| 3A                    | 19G20.0322N99G11.5638W | 2B28h         | 24°   | A             | 2B05h         | 3°   | C             | 21°  | 15°  |
| FA                    | 19G20.0330N99G11.5537W | 2B05h         | 3°    | A             | 2B05h         | 3°   | E             | 174° | 90°  |
| Prueba 2, Recorrido B |                        |               |       |               |               |      |               |      |      |
| Punto                 | Coordenadas            | Dir. $\gamma$ | Gamma | Caso $\gamma$ | Dir. $\alpha$ | Alfa | Caso $\alpha$ | Beta | Giro |
| IB                    | 19G20.0132N99G11.5775W |               |       |               |               |      |               |      |      |
| 2B                    | 19G20.0276N99G11.5760W | 2B8Ch         | 84°   | A             | 2B46h         | 15°  | C             | 69°  | 75°  |
| 3B                    | 19G20.0319N99G11.5629W | 2B28h         | 24°   | A             | 2B05h         | 3°   | C             | 21°  | 15°  |
| FB                    | 19G20.0330N99G11.5520W | 2B05h         | 3°    | A             | 2B05h         | 3°   | E             | 174° | 90°  |

Tabla 7.2 Hoja de Datos de la Prueba 2



El primer paso del análisis fue la comparación entre los puntos de los recorridos A y B, para comprobar que los recorridos fueran similares. Por ejemplo, entre los puntos 2A y 2B, la diferencia fue de 0.0002 minutos en la latitud y de 0.0001 minutos en la longitud. Al transformar estas diferencias a grados, como son muy pequeñas, el programa las elimina y considera que ambos puntos son idénticos. Por este motivo, en ambos puntos se obtuvieron los mismos datos de direcciones, ángulos, casos de orientación y ángulo de giro del motor de dirección. Esta situación se repite para el resto de los puntos de ambas trayectorias.

El segundo paso del análisis fue la verificación de la correspondencia entre las direcciones de memoria, los ángulos desplegados y valores de las tangentes calculadas para cada par de puntos. Por ejemplo, entre los puntos 1A y 2A se calcula un ángulo de trayectoria y de  $81.3^\circ$ , como este valor es mayor que  $81^\circ$ , le corresponde el valor de  $84^\circ$  en la tabla de tangentes definida para el sistema. La dirección de la tangente de  $84^\circ$  es 2B8Ch tal como se observa en la tabla 7.2. De la misma manera, se verifican los valores del ángulo de orientación  $\alpha$  para cada par de puntos.

Como último paso del análisis, se corrobora que los casos de orientación reportados concuerden con las posiciones registradas y que el ángulo de corrección  $\beta$  haya sido correctamente calculado. Es importante mencionar que en los puntos 1A e IB, por ser los puntos de inicio de la trayectoria, no existen los datos correspondientes, pues no hay referencia de donde obtenerlos. En los puntos FA y FB de la tabla 7.2, que son los puntos finales de cada recorrido (donde se cumplió el límite de acercamiento al punto de destino), los datos sombreados no deben considerarse dentro del análisis de la trayectoria pues en estos puntos el móvil se ha detenido. Además, la distancia del móvil al punto final no es suficiente para garantizar el cálculo correcto de estos datos.

De manera similar, se realizaron los análisis del resto de las pruebas concluyendo que el comportamiento del sistema es el correcto, mostrando buen desempeño en todos los recorridos. Puede comentarse que en los recorridos de las pruebas 1 y 2 (figuras 7.8 y 7.9) se presentan diferencias muy pequeñas, mientras que en los recorridos de las pruebas 3 y 4 (figuras 7.10 y 7.11) las diferencias son mayores. Sin embargo, estas discrepancias no tienen relevancia en la evaluación del sistema, pues se deben a la imprecisión de las desviaciones simuladas en la dirección del móvil y a la variación en la velocidad de su desplazamiento entre cada recorrido, no a los comandos de control procesados por el sistema.

### **7.3.2 Prueba De Verificación Del Límite De Acercamiento**

El procedimiento para la realización de estas pruebas consistió en registrar la coordenada de un punto que sería el de destino y luego desplazar al móvil desde un punto cualquiera hacia él. El sistema debía detener el motor de propulsión del móvil cuando se cumpliera la condición del límite de acercamiento definido. Repitiendo el procedimiento varias veces, se obtuvo un conjunto de valores de distancia en que el sistema cumplió con dicha condición. Para un límite de acercamiento definido de 9 (m) el conjunto de datos es el que muestra la tabla 7.3.

| Prueba | Distancia | Prueba   | Distancia |
|--------|-----------|----------|-----------|
| 1      | 2.7       | 9        | 4.9       |
| 2      | 3.9       | 10       | 6.4       |
| 3      | 6.2       | 11       | 4.7       |
| 4      | 6.3       | 12       | 4.6       |
| 5      | 1.9       | 13       | 8.8       |
| 6      | 1.9       | 14       | 12.4      |
| 7      | 2.1       | 15       | 16.9      |
| 8      | 3.4       | Promedio | 5.81      |

Tabla 7.3 Datos para Limite de Acercamiento de 9(m)

Como puede observarse en la tabla 7.3, el rango de valores de distancia para los que se cumplió la condición del límite de acercamiento es bastante grande, desde 1.9 (m) hasta 16.9 (m). Debe mencionarse que la magnitud de la distancia en la que se cumple el límite de acercamiento es variable y está sujeta a la dirección con la que se aproxime el móvil al punto de destino, pues el área definida por el límite de acercamiento es un cuadrilátero. La figura 7.12 muestra el área alrededor de un punto conocido demarcada por medio del límite de acercamiento de 9 (m).

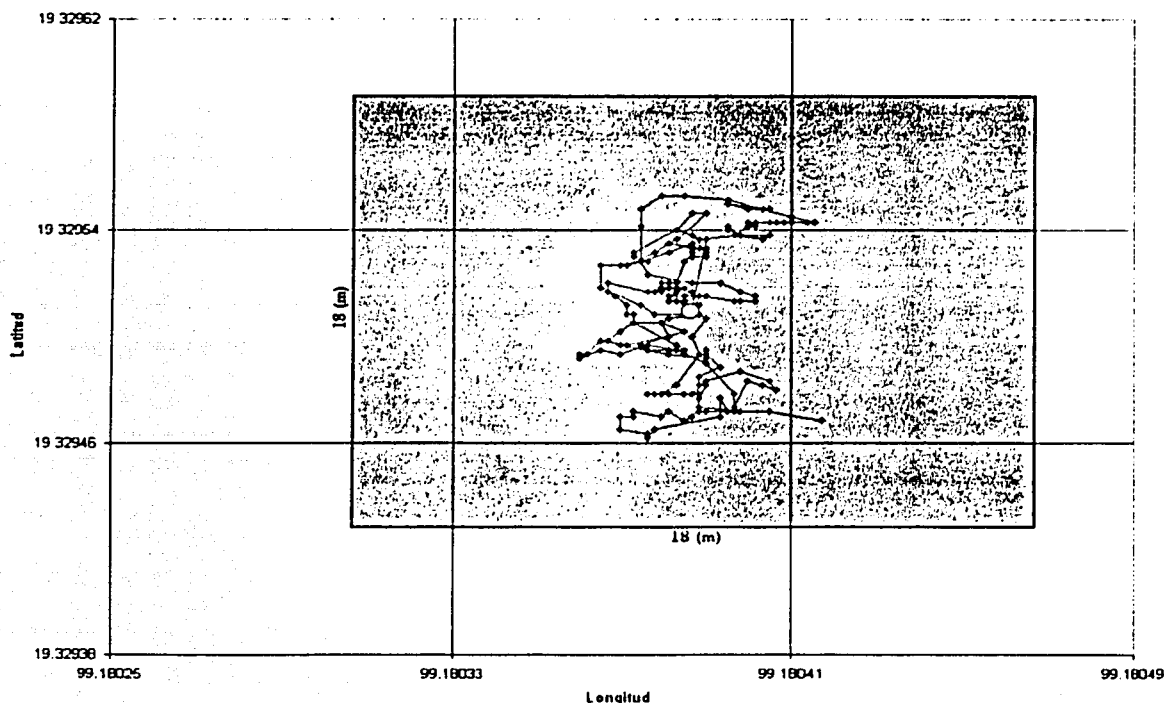


Figura 7.12 Área del Limite de Acercamiento de 9 (m)

En la figura 7.12 se aprecia un conjunto de puntos que representan una serie de 1000 lecturas del receptor GPS tomadas sobre una posición fija. Se graficaron 200 de estas 1000 lecturas, es decir, se omiten cinco lecturas intermedias por cada lectura graficada. De esta forma puede observarse el comportamiento del receptor GPS. En el centro de la gráfica aparece un pequeño círculo blanco para destacar el promedio de todas las lecturas del receptor GPS.

El área sombreada en la figura 7.12 resulta de marcar el límite de acercamiento: son 9 (m) a cada lado de la posición promedio sobre el eje de longitud y 9 (m) a cada lado sobre el eje de latitud. El resultado es un cuadrilátero de 18 (m) de lado alrededor del valor promedio.

Otro factor que afecta la distancia a la que el móvil se detiene del punto de destino es la precisión del receptor GPS. Para la figura 7.2 se calcularon las diferencias en latitud y longitud entre el valor promedio y el mayor de los valores registrados en cada caso. Para la longitud, la diferencia calculada fue de  $4.38 \text{ E-}05$  grados o 4.87 (m). Para la latitud, el valor de la diferencia fue de  $3.05 \text{ E-}05$  grados o 3.20 (m). La resultante de estas dos diferencias es de 5.83 (m), este es el valor de la desviación del receptor GPS calculada en un solo sentido. Para considerar el otro sentido deben agregarse otros 5.83 (m), con lo que la precisión total del receptor GPS es de 11.66 (m).

Finalmente, la distancia a la que el móvil considera que alcanzó su punto destino también será afectada dependiendo de la etapa del programa de control en que se encuentre el sistema, ya que existe una parte del programa en la que el móvil avanza sin comparar su posición.

A pesar de las variaciones en la distancia de acercamiento al punto de destino, el móvil se detuvo en todas las pruebas realizadas dentro de un valor aceptable de radio de acercamiento, dadas las aplicaciones para las que se diseñó el sistema.

**8****Conclusiones**

El sistema de control de navegación diseñado garantiza que, dados un tiempo de muestreo y un límite de radio de acercamiento calculados de acuerdo con la precisión del sistema satelital GPS, y contando con la disponibilidad de la señal proveniente del mismo, un vehículo marítimo que se desplace a través de un medio con pocas perturbaciones tendrá la capacidad de generar correcciones a su trayectoria, basándose únicamente en su posición, hasta alcanzar un punto de destino. Adicionalmente, ante la presencia de algún obstáculo en su trayectoria, el vehículo modificará su dirección de desplazamiento sin tomar en cuenta su posición para evitar una colisión con el mismo, esto significa que la evasión de obstáculos tiene prioridad sobre la corrección de trayectoria hacia el punto de destino.

El sistema desarrollado tiene la capacidad de adaptarse a las características de una amplia gama de vehículos marítimos que utilicen para su desplazamiento una propela y un timón, y por tanto, puede ser usado en diversas aplicaciones.

Al disponer de sistemas de navegación autónoma se deben aprovechar este tipo de medios en ocasiones en las que la utilización de vehículos tripulados puede representar peligro para el ser humano, como es el caso de rescates en zonas peligrosas, ante fugas químicas o radioactivas, en la extinción de incendios, en búsqueda y rescate de accidentados, o en reconocimiento de zonas costeras desconocidas.

Se ha pensado en algunas de las aplicaciones específicas en las que se podría utilizar el sistema de navegación presentado. Por ejemplo, si se quiere realizar un estudio de contaminación en el mar se puede enviar un vehículo, hacia una zona de interés y tomar las muestras necesarias. Durante el tiempo del estudio, el móvil cambiará su posición debido a las corrientes marinas que lo arrastran, cubriendo toda el área de investigación; posteriormente, se actualizarán las coordenadas del punto de destino del móvil para que regrese a su punto de partida y se lleve a cabo el análisis de las muestras recopiladas.

Una aplicación similar es cuando se quieren recuperar las radiobalizas marinas para su mantenimiento, o boyas que sirven a alguna aplicación y se encuentran a la deriva en el mar. El sistema de navegación propuesto podría instalarse en conjunción con estos equipos de modo que fuera posible la movilización automática de los mismos hacia un punto de concentración cercano a las costas para luego ser recuperados por conjunto de manera sencilla.

Otra aplicación está dirigida hacia la vigilancia de los puertos, las costas, o bien en mar abierto. En esta aplicación se equipa a un vehículo con cámaras de video para vigilar los lugares en los que sea necesario. Se definirá una ruta de monitoreo entre una base localizada en un puerto y un punto dentro del mar. El vehículo recorrería esta trayectoria monitoreando la actividad presente en esa región y las imágenes obtenidas serían transmitidas hacia la base en el puerto donde los elementos de seguridad calificarían las actividades y en caso de encontrar alguna actividad ilícita se mandaría a la guardia costera para la detención de los implicados. Una vez que el vehículo autónomo de vigilancia completa su ruta, se actualizarían sus coordenadas de destino para iniciar el regreso de la unidad hacia el puerto.

Una aplicación de índole militar puede ser el reconocimiento de áreas donde se piensa que existe peligro y no se quiera mandar equipo humano para verificar. Desde alguna embarcación que quisiera llegar a un puerto donde hay posible actividad hostil, puede enviarse un vehículo con el equipo necesario para determinar si es seguro desembarcar, de modo que se evita el exponer a las tropas a algún ataque.

Existen varios aspectos que pueden ser modificados para mejorar el desempeño del sistema, o para adaptarlo a ambientes con mayor número de variables que aquellas consideradas en este proyecto. Por ejemplo, pueden incluirse mecanismos auxiliares de posicionamiento que mejoren la precisión de la localización del móvil, como la utilización de GPS diferencial u otro sistema de aumento de precisión (WAAS, EGNOS, etc.). Asimismo, puede implementarse un subsistema de sensores con más elementos, más alcance, o un algoritmo de decisión más versátil y complejo. Una forma de optimizar el algoritmo de control del móvil es utilizando dispositivos que ayuden al cálculo de la trayectoria del móvil, tal es el caso de una brújula electrónica que aporta información sobre la orientación del vehículo de forma instantánea.

La implementación, por ejemplo, de un canal de radiocomunicación con el móvil, sería de gran ayuda para muchas aplicaciones en las que se deba actualizar la coordenada de destino del vehículo en forma remota.

Pese a las muchas mejoras que pueden lograrse sobre el sistema presentado, con este proyecto se aporta un conjunto de experiencias suficientes para establecer un modelo base sobre el cual se puede desarrollar la nueva tecnología enfocada hacia el diseño de sistemas similares. La materialización del sistema propuesto se logra a un costo bastante bajo, pues utiliza elementos sencillos y de fácil implementación. Esto lo pone al alcance de la mayoría de las instituciones educativas y de investigación, así como de particulares que necesiten disponer del sistema. Además, se desarrollaron subrutinas de software muy útiles y versátiles para sistemas de control similares al que se presentó en este escrito.

## APÉNDICE

|                                 |     |
|---------------------------------|-----|
| 1. Programa MAIN.ASM            | 94  |
| 1.1. Rutina MAIN                | 94  |
| 1.2. Subrutina DRDC             | 95  |
| 2. Programa 68HC11E9.INC        | 95  |
| 3. Programa GPS.ASM             | 96  |
| 3.1. Subrutina RECIBEI          | 96  |
| 3.2. Subrutina COORGPS          | 99  |
| 3.3. Subrutina GDCOORD          | 103 |
| 3.4. Subrutina DESP             | 104 |
| 4. Programa TAB.ASM             | 104 |
| 4.1. Subrutina INSAINI          | 105 |
| 4.2. Subrutina GAMA             | 105 |
| 4.3. Subrutina LLENAG           | 106 |
| 4.4. Subrutina BETA             | 107 |
| 5. Programa PALFA.ASM           | 109 |
| 5.1. Subrutina ALFA             | 109 |
| 5.2. Subrutina LLENAA           | 110 |
| 5.3. Subrutina CHKTAN           | 110 |
| 6. Programa MOTOR.ASM           | 112 |
| 6.1. Subrutina MOTORP           | 112 |
| 6.2. ISR de motor de propulsión | 112 |
| 6.3. Subrutina PASOS            | 113 |
| 6.4. Subrutina NUMERO           | 115 |
| 6.5. Subrutina ESPERA           | 115 |
| 7. Programa SENSOR.ASM          | 116 |
| 7.1. Subrutina SENSOR           | 116 |
| 7.2. ISR de sensores            | 116 |
| 8. Programa COMPARA.ASM         | 117 |
| 8.1. Subrutina LLEGA            | 117 |
| 9. Programa TABLA2.ASM          | 118 |
| 9.1. Subrutina DOTAB            | 119 |
| 10. Programa GD1.ASM            | 122 |
| 10.1. Subrutina GDCOORD         | 122 |

Nota: en este escrito no se presenta el listado de las subrutinas del paquete para aritmética de punto flotante (PNT.ASM) desarrolladas por Motorola, 1987 que se utilizaron en el programa de control. Se utilizó este paquete revisado por el Ing. José Antonio Arredondo Garza, del Depto. de Control de la Facultad de Ingeniería de la UNAM. Revisión de 1996.

1. MAIN.ASMPrograma principal que incluye todas las subrutinas  
utilizadas y controla su secuencia de ejecución

```

$INCLUDE "68HC11E9.INC" ;Contiene la definición de puertos del micro.
$INCLUDE "GPS.ASM"      ;Contiene subrutinas COORGPS y RECIBEL.
$INCLUDE "TAB.ASM"     ;Contiene ALFA, GAMA, BETA e INSAINI.
$INCLUDE "PNT.ASM"     ;Contiene las subrutinas numéricas de pto. flotante.
$INCLUDE "MOTOR.ASM"   ;Contiene subrutina MOTORP.
$INCLUDE "SENSOR.ASM"  ;Contiene subrutina SENSOR.
$INCLUDE "COMPARA.ASM" ;Contiene subrutina LLEGA.
$INCLUDE "TABLA2.ASM"  ;Contiene subrutina DOTAB.
$INLCUDE "GD1.ASM"    ;Contiene subrutina GDCOOR1.

```

```

CONT      EQU  $1040      ;Contador de lecturas de comparación

```

**1.1. RUTINA MAIN, CONTIENE LA SECUENCIA DE EJECUCIÓN DE LAS SUBRUTINAS  
PARA CONTROLAR EL SISTEMA DE NAVEGACIÓN**

```

MAIN      ORG  $1041      ;Dir. inicial del programa principal.
          JSR  DOTAB      ;Realiza la tabla de valores tangentes.
          JSR  RECIBEL    ;Espera coordenadas finales.
          JSR  MOTORP     ;Enciende el motor de propulsión.
          JSR  SENSOR     ;Enciende sensores.
          JSR  COORGPS    ;Obtiene coordenadas del GPS.
          JSR  INSAINI    ;Transfiere coordenadas instantáneas a iniciales.
OT        JSR  COORGPS    ;Obtiene nuevas coordenadas del GPS.
          JSR  LLEGA      ;Verifica si ha llegado al punto final.
          JSR  GAMA       ;Calcula el ángulo de trayectoria GAMMA.
          JSR  ALFA       ;Calcula el ángulo de dirección ALFA.
          JSR  BETA       ;Calcula ángulo de corrección BETA.
          JSR  NUMERO     ;Calcula número de pasos para mover motor dirección.
          JSR  PASOS      ;Mueve el motor de dirección.
          JSR  INSAINI    ;Transfiere coordenadas instantáneas a iniciales.
          LDAA #05        ;Número de comparaciones intermedias con el límite de
          STAA CONT      ;acercamiento, comparaciones entre correcciones.
CICLO     LDX  #$01CC     ;Ciclo en el que se realizan las comparaciones
N2        LDY  #$0CA0     ;del límite de acercamiento.
N1        DEY
          BNE  N1
          DEX
          BNE  N2
          JSR  COORGPS
          JSR  LLEGA
          LDAA CONT
          DECA
          STAA CONT
          BNE  CICLO      ;Si la condición del límite de acercamiento no se cumple
          BRA  OT         ;toma lectura para calcular corrección de trayectoria.
                          ;Repite ciclo OT hasta que el móvil llegue a su destino.

```

## 1.2. SUBROUTINA DRDC, DESPLIEGA EL REGISTRO DE CONTROL DE CASO DE ORIENTACIÓN

```

ORG $10CE ;Con esta subrutina se despliega el registro de control
DRDC LDX #RCTRL ;que ayuda a verificar el caso de orientación con el que
LDY #RCTRL ;se desplaza el vehículo.
LDAB #$01 ;La subrutina DESP despliega en pantalla el contenido de
ABY ;los registros de memoria que se encuentren entre la dir.
JSR DESP ;de X y la de Y.
RTS

```

FIN DE SUBROUTINA DRDC

### CONSTANTES EN EL PROGRAMA MAIN.ASM

```

ORG $1090 ;Este conjunto de datos limpia los registros
LIMPIA: FDB $2020,$2020 ;de memoria que utilizan los programas
FDB $2020,$2020 ;GPS.ASM y TAB.ASM
FDB $2020,$2020
FDB $2020,$2020

ORG $10A0 ;Tabla de estados para mover el motor de dirección
MXX: FDB $0102,$0408 ;contiene una serie de números (1,2,4,8) que
FDB $0102,$0408 ;indican el giro del motor de pasos.
FDB $0102,$0408 ;Estos datos se mandan al puerto B.
FDB $0102,$0408
FDB $0102,$0408
FDB $0102,$0408
FDB $0102,$0408
FDB $0102,$0408
FDB $0102,$0408
END

```

## 2. 68HC11E9.INC

En este programa se definen los registros del microprocesador M68HC11E9

```

PORTB EQU $1004 ;Puerto B que controla al motor de dirección.
PORTC EQU $1003 ;Puerto C en que se recibe la señal de los sensores.
PORTCL EQU $1005 ;Registro que guarda el último edo. del puerto C.
DDRC EQU $1007 ;Registro de control para el puerto C.
OPTION EQU $1039 ;Para controlar interrupciones.
PORTD EQU $1008 ;Puerto D de comunicación serial.
BAUD EQU $102B ;Registro de control de baudaje.
SCCR1 EQU $102C ;Registro de comunicación serial 1.
SCCR2 EQU $102D ;Registro de comunicación serial 2.
SCSR EQU $102E ;Registro de comunicación serial.
SCDR EQU $102F ;Registro de comunicación serial.
TCTL1 EQU $1020 ;Registro de reloj interno.
PACTL EQU $1026 ;Registro del puerto A.
TMSK1 EQU $1022 ;Registro para reloj interno 1.
TFLG1 EQU $1023 ;Registro para reloj interno bandera 1.
TCNT EQU $100E ;Registro para reloj interno contador.
TOC3 EQU $101A ;Registro de salida de comparación 3.

```



## 3. GPS.ASM

Este programa contiene las funciones para manejar la información del GPS.

DECLARACION DEL ENCABEZADO DE LA CADENA GPS UTILIZADA: \$GPGGA,<CR><LF>

|   |     |        |  |
|---|-----|--------|--|
| CHR1  | EQU | \$3000 | ;Guarda la Primera 'G' de la cadena hex= 47.   |
| CHR2  | EQU | \$3001 | ;Guarda la letra 'P' de la cadena 50.          |
| CHR3  | EQU | \$3002 | ;Guarda la Segunda 'G' de la cadena 47.        |
| CHR4  | EQU | \$3003 | ;Guarda la Tercera 'G' de la cadena 47.        |
| CR  | EQU | \$3004 | ;Guarda el Carriage Return de la cadena 0D.    |
| LF  | EQU | \$3005 | ;Guarda el Line Feed de la cadena 0B.          |
| INIGPGGD  | EQU | \$3006 | ;Inicio sentencia GPGG, registro doble.        |
| FINGPGGD  | EQU | \$3008 | ;Fin sentencia GPGG, registro doble.           |
| ITRAMAD   | EQU | \$300A | ;Inicio trama GPS, es un registro doble.       |
| FTRAMAD   | EQU | \$300C | ;Fin trama GPS, registro doble.                |
| NCDA  | EQU | \$300E | ;No. de caracteres definidos después de '*'. . |
| DITRAMAD  | EQU | \$3010 | ;Dirección base cadena 'GPGG' registro doble.  |
| RGEQUS  | EQU | \$3012 | ;Registro auxiliar X.                          |
| RGYE  | EQU | \$3014 | ;Registro auxiliar Y.                          |
| INITABD   | EQU | \$3016 | ;Inicio de la tabla de coordenadas.            |
| FINTABD   | EQU | \$3018 | ;Fin de la tabla de coordenadas.               |
| ;Memoria de \$3020 a \$3122 ocupada por la sentencia del GPGG |     |        |  |
| MIN1  | EQU | \$3023 | ;Dir de inicio de minutos de latitud.          |
| GRD2  | EQU | \$302B | ;Dir de inicio de grados de longitud.          |
| MIN2  | EQU | \$302F | ;Dir de inicio de minutos de longitud.         |
| LATFIN  | EQU | \$3124 | ;Va de \$0600-\$060A, son 12 caracts.          |
| LONFIN  | EQU | \$3126 | ;Va de \$060D-\$0617, son 12 caracts.          |
| LATINS  | EQU | \$3128 | ;Va de \$0619-\$0623, son 12 caracts.          |
| LONINS  | EQU | \$312A | ;Va de \$0628-\$0635, son 12 caracts.          |
| INICORF   | EQU | \$312C | ;Registro de inicio de coordenadas.            |
| FINCORF   | EQU | \$312E | ;Registro de fin de coordenadas.               |
| RESULT  | EQU | \$3130 | ;Dir donde se coloca resultado.                |

## 3.1. SUBROUTINA RECIBE1

DEFINICIÓN DE LAS DIRECCIONES DE VARIABLES A UTILIZAR

|         |      |         |   |
|---------|------|---------|---|
|         | ORG  | \$3200  | ;Dirección de inicio de programa.               |
| RECIBE1 | LDX  | #\$0600 |   |
|         | STX  | LATFIN  | ;Guarda dir. de latitud final.                  |
|         | LDX  | #\$060D |   |
|         | STX  | LONFIN  | ;Guarda dir. de longitud final.                 |
|         | LDX  | #\$0619 |   |
|         | STX  | LATINS  | ;Guarda dir. de latitud instantánea.            |
|         | LDX  | #\$0628 |   |
|         | STX  | LONINS  | ;Guarda dir. de longitud instantánea            |
|         | LDX  | #\$0703 |   |
|         | STX  | INICORF | ;Guarda dir. del inicio de coordenadas finales. |
|         | LDX  | #\$0716 |   |
|         | STX  | FINCORF | ;Guarda dir. del fin de coordenadas finales.    |
|         | LDAA | #\$20   |   |
|         | STAA | \$0700  | ;Guarda en dir. \$0700 un ESPACIO.              |
|         | LDAA | #\$0A   |   |
|         | STAA | \$0701  | ;Guarda en dir. \$0701 un LINE FEED.            |
|         | LDAA | #\$0D   |   |
|         | STAA | \$0702  | ;Guarda en dir. \$0702 un CARRIEGE RETURN.      |

## RECIBE LAS COORDENADAS FINALES (DESTINO) DESDE TECLADO

```

LDAA #S04 ;Habilita la recepción del micro cargando un 1
STAA SCCR2 ;en bit 2 (RE) de SCCR2 (SCI Control Register 2).
LDAA #S30 ;Especifica velocidad de recepción de 9600 bauds
STAA BAUD ;cargando un 1 en bits 4 y 5 (SCP0 y SCP1) de BAUD.
LDX INICORF ;Carga en X la dir. base para coord finales.
CHRDRF1 LDAA SCSR ;Lee el registro SCSR.
ANDA #S20 ;Verifica si el bit RDRF de SCSR está en alto y
BEQ CHRDRF1 ;hasta que se levante lo esta checando.
LDAA SCSR ;Lee registro SCSR y al leer enseguida el registro
LDAA SCDR ;SCDR borra bandera RDRF. En SCDR recibe un carácter
STAA 0,X ;que guarda a partir de la dir. INICORF.
INX ;Incrementa X para tener la sig. dir. de memoria.
CPX FINCORF ;Compara X con dir. FINCORF, (que sean 18 caracteres).
BEQ DESP1 ;Si son 18 caracteres va a desplegarlos.
JMP CHRDRF1 ;Si aun NO son 18 caracteres vuelve a recibir.

DESP1 LDX INICORF ;Carga en X la dir. INICORF.
LDY FINCORF ;Carga en Y la dir. FINCORF.
JSR DESP ;Va a subrutina DESP (despliega 18 caracteres).

```

## CONFIRMA SI LAS COORDENADAS DE DESTINO SON CORRECTAS

```

LDAA #S04 ;Habilita la recepción del micro cargando un 1
STAA SCCR2 ;en bit 2 (RE) de SCCR2 (SCI Control Register 2).
LDAA #S30 ;Especifica velocidad de recepción de 9600 bauds
STAA BAUD ;cargando un 1 en bits 4 y 5 (SCP0 y SCP1) de BAUD.
CHKRDRF LDAA SCSR ;Lee el registro SCSR.
ANDA #S20 ;Verifica si el bit RDRF de SCSR está en alto y
BEQ CHKRDRF ;hasta que se levante lo esta checando.
LDAA SCSR ;Lee registro SCSR y al leer en seguida el registro
LDAA SCDR ;SCDR borra bandera RDRF. En SCDR recibe un carácter
CMPA #S53 ;y compara con 'S' para confirmar que las coordenadas
BEQ CAMBIO1 ;tomadas son correctas. Si es así, va a CAMBIO1 y
JMP RECIBEL ;si NO son correctas regresa a RECIBEL.

```

## REVISA LATITUD DE COORD FINALES, SI ES NORTE O SUR

```

CAMBIO1 LDX INICORF ;Carga en X la dir. inicial de coord. finales y en A
LDAA 08,X ;lo de dir. INICORF+8, carácter que indica 'N' o 'S'.
CMPA #S4E ;Compara A con el carácter 'N' y si es igual
BEQ NORTE1 ;va a NORTE1.
JMP SUR1 ;Si la comparación no es igual, va a SUR1

```

## SI LA LATITUD ES NORTE, CAMBIA SISTEMA Y GUARDA LATFIN

```

NORTE1 LDX INICORF ;Carga en X la dir inicial de coord. finales
LDY #NOVENT ;Carga en Y la dir de un 90 en ascii
JSR SUMA ;Cambia sistema: LATFIN=latitudN+90, queda en FPACC1
LDX #RESULT ;Carga en X dir inicial para guardar coord en ascii
JSR FLTASC ;y salta a subrutina para pasar deflt. a ascii.
LDX #RESULT ;Carga en X dir base donde est el resultado y
LDY LATFIN ;en Y dir donde se guardar como LATFIN (ascii).
JSR GDCOOR ;Va a guardar coordenada de RESULT a LATFIN.
JMP LONG1 ;Va a función para cambiar de sistema LONGITUD.

```

## SI LA LATITUD ES SUR, CAMBIA DE SISTEMA Y GUARDA LATFIN

```

SUR1      LDX      #NOVENT  ;Carga en X la dir. de un 90 en ascii.
          LDY      INICORF  ;Carga en Y la dir. inicial de coord. finales.
          JSR      RESTA    ;Cambia sistema: LATFIN=90-LatitudS, queda en FPACCl
          LDX      #RESULT  ;Carga en X dir. inicial para guardar coord. en ascii
          JSR      FLTASC   ;y salta a subrutina para pasar de flot. a ascii.
          LDX      #RESULT  ;Carga en X la dir. base donde está el resultado y
          LDY      LATFIN   ;en Y dir. donde se guardará como LATFIN (ascii).
          JSR      GDCOOR   ;Va a guardar coordenada de RESULT a LATFIN.
          JMP      LONG1    ;Va a función para cambiar de sistema LONGITUD.

```

## REvisa LONGITUD DE COORD FINALES, SI ES ESTE U OESTE

```

LONG1     LDX      INICORF  ;Carga en X dir. inicial de coord. finales y en A
          LDAA     12,X      ;lo de dir. INICORF+12, carácter que indica 'E' o 'W'.
          CMPA    #$45      ;Compara A con el carácter 'E', y
          BEQ     ESTE1     ;si es igual va a ESTE1.
          JMP     OESTE1    ;Si la comparación NO fue igual va a OESTE1.

```

## SI LA LONGITUD ES ESTE, CAMBIA DE SISTEMA Y GUARDA LONFIN

```

ESTE1     LDX      #CTOCH   ;Carga en X la dir. de un 180 en ascii.
          LDY      INICORF  ;Carga en Y la dir. inicial de coord. finales.
          LDAB    #$09      ;Carga en B un 9h para sumar a la dir. en Y.
          ABY     ;Suma B+Y y queda en Y la dir. $070C (longitud).
          JSR     SUMA      ;Cambia sistema: LONFIN=180+LongitudE, queda en FPACCl
          LDX      #RESULT  ;Carga en X dir. inicial para guardar coord. en ascii.
          JSR     FLTASC   ;y salta a subrutina para pasar de flot. a ascii.
          LDX      #RESULT  ;Carga en X dir. base donde está el resultado y
          LDY     LONFIN   ;en Y dir. donde se guardar como LONFIN (ascii).
          JSR     GDCOOR   ;Va a guardar coordenada de RESULT a LONFIN.
          JMP     DESPFIN  ;Salta a subrutina para desplegar coord. finales.

```

## SI LA LONGITUD ES OESTE, CAMBIA DE SISTEMA Y GUARDA LONFIN

```

OESTE1    LDX      #CTOCH   ;Carga en X la dir. de un 180 en ascii.
          LDY      INICORF  ;Carga en Y la dir. inicial de coord. finales.
          LDAB    #$09      ;Carga en B un 9h para sumar a la dir. en Y
          ABY     ;Suma B+Y y queda en Y la dir. $070C (longitud)
          JSR     RESTA    ;Cambia sistema: LONFIN=180-LongitudW, queda en FPACCl
          LDX      #RESULT  ;Carga en X dir. inicial para guardar coord. en ascii.
          JSR     FLTASC   ;y salta a subrutina para pasar de flot. a ascii.
          LDX      #RESULT  ;Carga en X dir. base donde está el resultado y
          LDY     LONFIN   ;en Y dir. donde se guardar como LONFIN (ascii).
          JSR     GDCOOR   ;Va a guardar coordenada de RESULT a LONFIN
DESPFIN   LDX      LATFIN   ;Despliega las coordenadas finales
          LDY     LATFIN   ;en sistema absoluto.
          LDAB    #$0C      ;Carga las direcciones de inicio y fin de
          ABY     ;la información a desplegar en X y Y respectivamente.

```

```

JSR   DESP   ;y manda a subrutina DESP.
LDX   LONFIN ;Lo mismo para la longitud.
LDY   LONFIN
LDAB  #$0C   ;El número de registros de memoria desplegados es 0Ch.
ABY
JSR   DESP   ;Subrutina despliega.
RTS    ;Termina subrutina RECIBEL, regresa a MAIN.

```

FIN DE SUBROUTINA RECIBEL, REGRESA A MAIN

### 3.2. SUBROUTINA COORGPS, TOMA LECTURA DE POSICIÓN DEL RECEPTOR GPS.

GUARDA CARACTERES QUE SIRVEN PARA IDENTIFICAR SENTENCIA 'GPGG'

```

COORGPS  LDAA  #$47
          STAA  CHR1   ;La localidad CHR1 de memoria contiene 'G'.
          STAA  CHR3   ;CHR3 contiene 'G'.
          STAA  CHR4   ;CHR4 contiene 'G'.
          LDAA  #$50
          STAA  CHR2   ;CHR2 contiene 'P'.
          LDAA  #$0D
          STAA  CR     ;CR contiene un Retorno de Carro.
          LDAA  #$0A
          STAA  LF     ;LF contiene un Linefeed.
          LDD   #$3020 ;Carga en D dir. inicial para trama GPS
          STD   ITRAMAD ;y guarda este valor en InicioTRAMADoble.
          LDD   #$3122 ;Carga en D dir. final para trama GPS
          STD   FTRAMAD ;y guarda este valor en FinTRAMADoble.
          LDD   #$0005 ;Guarda en NCDA 5h, que indica el No. de caracteres
          STD   NCDA   ;definidos luego de un '*' en la trama.
          LDD   #$3000 ;Carga en D dir. inicial de la trama 'GPGG' capturada
          STD   DITRAMAD ;y la guarda en DirInicioTRAMADoble.
          LDX   #$3020 ;Carga en X dir. inicio para crear tabla de
          STX   INITABD ;coordenadas y lo guarda en INICIOTABlaDoble.
          LDX   #$303A ;Carga en X dir. fin para crear tabla de coordenadas
          STX   FINTABD ;y lo guarda en FINTABlaDoble.

```

RECIBE Y ALMACENA TRAMA GPS COMPLETA, DE ITRAMAD A FTRAMAD

```

RECIBE   LDX   ITRAMAD ;Inicia X en la dir. base para la trama GPS
          LDAA  #$04
          STAA  SCCR2   ;Habilita recepción del microprocesador.
          LDAA  #$31
          STAA  BAUD    ;Define velocidad de transmisión a 9600 bauds.
          LDAA  SCSR    ;Lee registro SCSR y verifica si
          ANDA  #$20    ;la bandera RDRF esta en alto.
          BEQ  CHRDRF   ;Continua checando hasta que se levante RDRF.
          LDAA  SCSR    ;Borra bandera
          LDAA  SCDR    ;y carga en A el caracter recibido.
          STAA  0,X     ;Guarda caracter en memoria, inicia en ITRAMAD, $3020
          INX          ;Incrementa X para tener sig. dir. de memoria.
          CPX   FTRAMAD ;Compara X con FTRAMAD, $3122 deben ser 228 caracteres.
          BEQ  SELEC    ;Si son 228 caract. termina guardar y va a seleccionar.
          JMP  CHRDRF   ;Si aun NO son 228 caracteres vuelve a recibir otro.

```

TECIS CON  
FALLA DE ORIGEN

## IDENTIFICA SENTENCIA GPGG Y GUARDA DIR DE INICIO Y FIN DE LA MISMA

|         |       |          |  |
|---------|-------|----------|--|
| SELEC   |       | ;CLI     |  |
| PRIMERO | LDX   | ITRAMAD  | ;Carga en X dir. inicial de trama GPS recibida.        |
|         | LDY   | DITRAMAD | ;Se carga la dir. de inicio del conjunto 'GPGG'        |
|         | LDAA  | 0,Y      | ;Carga en A primer carácter (G) de 'GPGG'.             |
|         | CMPA  | 0,X      | ;Compara con primer carácter recibido de trama GPS.    |
|         | BEQ   | LUGAR    | ;Si son iguales va a revisar en donde lo encontró.     |
| LUGAR   | INX   |          | ;Si NO son iguales incrementa X y continua buscando,   |
|         | JMP   | PRIMERO  | ;vuelve a comparar los caracteres.                     |
|         | XGDX  |          | ;Cambia X con D, en X está dir. donde encontró 'G'.    |
|         | SUBD  | #\$30C0  | ;Resta: D-30C0h, si el resultado es mayor o igual,     |
|         | BHS   | RECIBE   | ;la cadena GPS fue mal guardada y pedirá otra.         |
| SEGUNDO | ADDD  | #\$30C0  | ;Suma a D el valor que se había restado.               |
|         | XGDX  |          | ;Se intercambia D con X para recuperar X.              |
|         | INX   |          | ;Incrementa X, es el carácter después de una 'G'.      |
|         | INY   |          | ;Incrementa Y, tiene segundo carácter de 'GPGG', P.    |
|         | LDAA  | 0,Y      | ;Carga en A segundo carácter (P) de GPGG.              |
| TERCERO | CMPA  | 0,X      | ;Compara el carácter de X con 'P'.                     |
|         | BEQ   | TERCERO  | ;Si son iguales busca el tercer carácter de GPGG.      |
|         | JMP   | PRIMERO  | ;Si NO son iguales, buscará nuevamente la primera 'G'. |
|         | INX   |          | ;Incrementa X, es el carácter después de 'GP'.         |
|         | INY   |          | ;Incrementa Y, tiene tercer carácter de GPGG, G.       |
| CUARTO  | LDAA  | 0,Y      | ;Carga en A tercer carácter (G) de GPGG.               |
|         | CMPA  | 0,X      | ;Compara el carácter de X con G.                       |
|         | BEQ   | CUARTO   | ;Si son iguales busca el cuarto carácter de GPGG.      |
|         | JMP   | PRIMERO  | ;Si NO son iguales, buscará nuevamente la primera 'G'. |
|         | INX   |          | ;Incrementa X, es el carácter después de 'GPG'.        |
| RGRESAX | INY   |          | ;Incrementa Y, tiene cuarto carácter de GPGG, G.       |
|         | LDAA  | 0,Y      | ;Coloca en A cuarto carácter (G) de GPGG.              |
|         | CMPA  | 0,X      | ;Compara el carácter de X con G.                       |
|         | BEQ   | RGRESAX  | ;Si son iguales decrementa X hasta dir. de primera G.  |
|         | JMP   | PRIMERO  | ;Si NO son iguales comienza a buscar nuevamente.       |
| BSCAA   | DEX   |          | ;Decrementa X hasta dir. de la primera 'G'             |
|         | DEX   |          | ;que marca el principio de la sentencia encontrada.    |
|         | DEX   |          |  |
|         | STX   | INIGPGGD | ;Guarda dirección de inicio de sentencia GPGG.         |
|         | added | added    | added  |
| BSCAA   | added | added    | added  |
|         | added | added    | added  |
|         | added | added    | added  |
|         | added | added    | added  |
|         | added | added    | added  |
| GFGPGG  | added | added    | added  |
|         | added | added    | added  |
|         | added | added    | added  |
|         | added | added    | added  |
|         | added | added    | added  |

SEPARA LAT Y LONG DE SENTENCIA GPGG Y GUARDA EN TABLA DE INITABD A FINTABD,  
EL FORMATO ES: 19G19.7638N099G10.8300 W

|         |      |          |   |
|---------|------|----------|---|
| EXTRACC | LDX  | INIGPGGD | ;Carga en X dir. de inicio de sentencia GPGG.         |
|         | LDY  | INITABD  | ;Carga en Y dir. de inicio de tabla.                  |
|         | LDAB | #\$02    | ;B sirve como contador.                               |
| STLAT1  | LDAA | 11,X     | ;Carga en A el contenido de localidad INIGPGGD+17d,   |
|         | STAA | 0,Y      | ;es un dígito de latitud y se guarda en la dir. en Y. |

```

INX          ;Incrementa X, siguiente digito latitud.
INY          ;Incrementa Y, siguiente localidad de tabla.
DECB        ;Decrementa B, contador.
BNE STLAT1  ;Si B!=0 guarda otro digito Si B=0 termino el entero.
LDAA #\$47   ;Carga en A una G de 'Grados' (señalización).
STAA 0,Y    ;Guarda la G después del entero de latitud.
INY         ;Incrementa Y, siguiente localidad para tabla.
LDAB #\$07   ;Inicializa contador en B con 7d.
STLAT2     LDAA 11,X ;Carga en A contenido de localidad en X, (X+17d).
STAA 0,Y    ;Guarda digito de fracción de latitud en dir. de Y.
INX        ;Incrementa X, sig. digito de fracción de latitud.
INY        ;Incrementa Y, sig. localidad memoria para tabla.
DECB       ;Decrementa B, contador.
BNE STLAT2  ;Si B!=0 guarda otro digito Si B=0 termino fracc. Lat.
INX        ;Incrementa X, tiene el sig. carácter después de lat.
LDAA 11,X   ;Carga en A el carácter, es 'N' o 'S'.
STAA 0,Y    ;Guarda carácter 'N' o 'S' (Señalización).
INY        ;Incrementa Y, sig. localidad de memoria para tabla.
INX        ;Incrementa X 2 veces para llegar al primer digito
INX        ;de la longitud.
LDAB #\$03   ;Inicializa contador B con 3d.
STLON1     LDAA 11,X ;Carga en A contenido de localidad X (X+17d)
STAA 0,Y    ;Guarda digito de long en dir. de Y.
INX        ;Incrementa X, siguiente digito.
INY        ;Incrementa Y, siguiente localidad para tabla.
DECB       ;Decrementa B, contador.
BNE STLON1  ;Si B!=0 guarda otro digito. B=0 termino entero long.
LDAA #\$47   ;Carga en A carácter 'G' de 'Grados' (señalización).
STAA 0,Y    ;Guarda carácter G después de entero de longitud.
INY        ;Incrementa Y, sig. localidad de memoria para tabla.
LDAB #\$07   ;Inicializa contador otra vez para las fracc. de long.
STLON2     LDAA 11,X ;Carga en A contenido de localidad X (X+17d).
STAA 0,Y    ;Guarda digito de fracción de longitud en dir. de Y.
INX        ;Incrementa X, siguiente digito fracc. de longitud.
INY        ;Incrementa Y, siguiente localidad para tabla.
DECB       ;Decrementa B, contador.
BNE STLON2  ;Si B!=0 guarda otro digito Si B=0 termino fracc. long.
INX        ;Incrementa X, siguiente carácter.
LDAA #\$20   ;Carga en A un ESPACIO, (señalización).
STAA 0,Y    ;Guarda el Espacio después de la longitud.
INY        ;Incrementa Y, sig. localidad para tabla.
LDAA 11,X   ;Carga en A el carácter, es 'E' o 'W'.
STAA 0,Y    ;Guarda carácter E o W (señalización).
INY        ;Incrementa Y, sig. localidad de memoria para tabla.
LDAA CR     ;Carga en A un CARRIEGE RETURN (señalización)
STAA 0,Y    ;Guarda el Carriage Return.
INY        ;Incrementa Y, sig. localidad de memoria para tabla.
LDAA LF     ;Carga en A un LINE FEED (señalización).
STAA 0,Y    ;Guarda Line Feed, FIN de tabla de coord. FINTABD
LDX INITABD ;Despliega la cadena GPS, en formato de grados,
LDY FINTABD ;minutos y fracciones de minutos.
JSR DESP    ;Subrutina despliega.

```

## CONVIERTE MINUTOS DE LATITUD A DÉCIMAS DE GRADO Y SUMA A GRADOS

```

LATITUD  LDX  #LIMPIA      ;Limpia las localidades de memoria
          LDY  LATINS      ;Limpia las localidades de memoria
          JSR  GDCOOR      ;que utilizará en esta subrutina.
          LDX  #MIN1       ;Carga en X los minutos de latitud.
          LDY  #OPER2      ;Carga en Y un 60d, desde dir. OPER2.
          JSR  DIVISION    ;Divide minutos/60, obtiene décimas de grado.
          LDX  #RESULT     ;Carga en X dir. base para el resultado en ascii.
          JSR  FLTASC      ;Convierte de flot. a ascii y guarda en dir. de X.
          LDX  #RESULT     ;Carga en X dir. de operando de SUMA (décimas de grado)
          LDY  INITABD     ;Carga en Y dir. donde están los grados de latitud.
          JSR  SUMA        ;Las décimas de grado las suma a los grados.
          LDX  #RESULT     ;Carga en X dir. base para resultado en ascii.
          JSR  FLTASC      ;Convierte de flot. a ascii y guarda en dir. de X.

```

## REVISAS SI LA LATITUD ES NORTE O SUR Y LA CAMBIA AL SISTEMA X y Y

```

          LDX  INITABD     ;Carga en X dir. de inicio de tabla.
          LDAA OA,X       ;Carga en A el carácter N o S, dir. INITABD+10d.
          CMPA #$4E       ;Compara el carácter con 'N'.
          BEQ  NORTE      ;Si es N va a NORTE.
          JMP  SUR        ;Si no es N, va a SUR.
NORTE    LDX  #RESULT     ;Carga en X dir. de latitud en grados y décimas de gdo.
          LDY  #NOVENT    ;Carga en Y dir. de un 90d.
          JSR  SUMA       ;Cambia sistema: LATINS=LatitudN+90d.
          LDX  #RESULT     ;Carga dir. donde estará el resultado en ascii,
          JSR  FLTASC      ;convierte de flot. a ascii y guarda en dir. de X.
          LDX  #RESULT     ;Carga en X dir. del resultado en ascii y
          LDY  LATINS      ;en Y dir. donde se guardará como LATINS.
          JSR  GDCOOR      ;Guarda latitud, de RESULT A LATitudINSTantanea.
          JMP  LONG       ;Salta a procesar información de longitud.
SUR      LDX  #NOVENT    ;Carga en X dir. de un 90d.
          LDY  #RESULT     ;Carga en Y dir. de latitud en grados y décimas de gdo.
          JSR  RESTA      ;Cambia sistema: LATINS=90d-LatitudS.
          LDX  #RESULT     ;Carga dir. donde estará el resultado en ascii,
          JSR  FLTASC      ;convierte de flot. a ascii y guarda en dir. de X.
          LDX  #RESULT     ;Carga en X dir. del resultado en ascii y
          LDY  LATINS      ;en Y dir. donde se guardará como LATINS.
          JSR  GDCOOR      ;Guarda latitud, de RESULT a LATINS.

```

## CONVIERTE MINUTOS DE LONGITUD A DÉCIMAS DE GRADO Y SUMA A GRADOS

```

LONG     LDX  #LIMPIA      ;Limpia las localidades de memoria que
          LDY  LONINS      ;utiliza el programa.
          JSR  GDCOOR
          LDX  #MIN2       ;Carga en X los minutos de longitud.
          LDY  #OPER2      ;Carga en Y un 60d, desde dir. OPER2.
          JSR  DIVISION    ;Divide minutos/60, obtiene décimas de grado.
          LDX  #RESULT     ;Carga en X dir. base para el resultado en ascii.
          JSR  FLTASC      ;Convierte de flot. a ascii y guarda en dir. de X.
          LDX  #RESULT     ;Carga en X dir. de operando de SUMA (décimas de grado)
          LDY  #GRD2      ;Carga en Y dir. donde inician grados de longitud.
          JSR  SUMA        ;Las décimas de grado las suma a los grados.
          LDX  #RESULT     ;Carga en X dir. base para resultado en ascii.
          JSR  FLTASC      ;Convierte de flot. a ascii y guarda en dir. de X.

```

REVISA SI LA LONGITUD ES ESTE U OESTE Y LA CAMBIA AL SISTEMA X y Y

```

LDX  INITABD  ;Carga en X dir. de inicio de tabla
LDAA  17,X    ;Carga en A el carácter 'W' o 'E', dir. INITABD+23d.
CMPA  #$45   ;Compara el carácter con 'E'.
BEQ   ESTE   ;Si es E va a ESTE.
ESTE  JMP     OESTE ;Si no es E, va a OESTE.
LDX   #CTOCH ;Carga en X dir. de un 180d.
LDY   #RESULT ;Carga en Y dir. de longitud en grados y décimas gdo.
JSR   SUMA   ;Cambia sistema: LONINS=LongitudE+180d.
LDX   #RESULT ;Carga dir. donde estará resultado en ascii,
JSR   FLTASC ;convierte de flot. a ascii y guarda en dir. de X.
LDX   #RESULT ;Carga en X dir. del resultado en ascii y
LDY   LONINS ;en Y dir. donde se guardará como LONINS.
JSR   GDCOOR ;Guarda longitud, de RESULT a LONGitudINstantanea.
OESTE JMP     DESPINS ;Despliega coordenadas instantáneas.
LDX   #CTOCH ;Carga en X dir. de un 180d.
LDY   #RESULT ;Carga en Y dir. de latitud en grados y décimas gdo.
JSR   RESTA ;Cambia sistema: LONINS=180d-LongitudO
LDX   #RESULT ;Carga dir. donde estará resultado en ascii,
JSR   FLTASC ;convierte de flot. a ascii y guarda en dir. de X.
LDX   #RESULT ;Carga en X dir. del resultado en ascii y
LDY   LONINS ;en Y dir. donde se guardará como LONINS.
DESPINS JSR   GDCOOR ;Guarda longitud, de RESULT a LONINS.
LDX   LATINS ;Despliega Latitud Instantánea
LDY   LATINS ;en formato de grados y fracciones de grado.
LDAB  #$0C
ABY
JSR   DESP
LDX   LONINS ;Despliega Longitud Instantánea en formato
LDY   LONINS ;de grados y fracciones de grado.
LDAB  #$0C
ABY
JSR   DESP
RTS   ;Termina subrutina.

```

FIN DE SUBROUTINA COORGPS, REGRESA A MAIN

3.3. SUBROUTINA GDCOOR LLAMADA DESDE SUBROUTINA COORGPS,  
GUARDA COORDENADAS EN MEMORIA DEFINIDA.

```

GDCOOR LDAB  #$0C ;B sirve de contador.
GUARDA LDAA  0,X  ;Carga A con el contenido de la dir. en X.
STAA  0,Y  ;Guarda contenido de A en dir. de Y.
INX   ;Incrementa X, sig. carácter.
INY   ;Incrementa Y, siguiente localidad para guardar.
DECB  ;Decrementa B, contador.
BNE   GUARDA ;Si B!=0 guarda otro carácter Si B=0 terminó de guardar
RTS   ;coordenadas y regresa de subrutina.

```

FIN DE SUBROUTINA GDCOOR, REGRESA A COORGPS



### 3.4. SUBROUTINA DESP, DESPLIEGA EN PANTALLA CARACTERES SITUADOS ENTRE DOS LOCALIDADES INICIAL Y FINAL CARGADAS ANTES EN X Y Y.

```

DESP      LDAA  #S08
          STAA  SCCR2      ;Habilita transmisión del microprocesador.
          LDAA  #S30
          STAA  BAUD      ;Define velocidad de transmisión de 9600 baud.
          STY   RGYE      ;Guarda dir. de carácter final en variable RIGE.
CHKTDRE   LDAA  SCSR      ;Lee SCSR, TDRE está en 1 cuando SCDR esta vacío.
          BMI   TX2       ;TDRE es el primer bit de SCSR (bit de signo).
          JMP   CHKTDRE    ;Verifica bandera hasta que se vacíe TDR.
TX2       LDAA  0,X       ;Carga en A primer carácter de cadena seleccionada.
          LDAB  SCSR      ;Para borrar bandera TDRE lee de SCSR y
          STAA  SCDR      ;escribe en SCDR;
CHKTC     LDAA  SCSR      ;Lee SCSR para saber si se transmitió todo el carácter.
          ANDA  #S40      ;Compara bandera TransmitionComplete del SCSR,
          BEQ   CHKTC     ;checa bandera hasta que se levante.
          INX   X         ;Incrementa X para tener el siguiente carácter.
          CPX   RGYE      ;Compara con dir. final de último carácter a desplegar.
          BEQ   FIN       ;Si dir. es la final, termina,
          JMP   CHKTDRE    ;si no, transmite otro carácter.
FIN       RTS            ;Termina de desplegar, regreso de subrutina.

```

FIN DE SUBROUTINA DESP, REGRESA A RECIBE1

### OPERANDOS EN GPS.ASM, VALORES QUE SE MANIPULAN PARA CAMBIO DE SISTEMA

```

          ORG  $35A5
OPER2:   FDB  $3630,$2020 ;Es un 60 decimal en formato ASCII.
          FDB  $0A0D

          ORG  $35AB
CTOCH:   FDB  $3138,$3020 ;Es un 180 decimal en formato ASCII.
          FDB  $2020

          ORG  $35B1
NOVENT:  FDB  $3930,$2020 ;Es un 90 decimal en formato ASCII.
          END

```

### 4. TAB.ASM

Programa que realiza el cálculo de los ángulos Gamma y Beta.

### DECLARACIÓN DE VARIABLES

```

LATINI   EQU   $35B6      ;Dos localidades, latitud inicial.
LONINI   EQU   $35B8      ;Dos localidades, longitud inicial.
DELTALONG EQU  $35BA      ;Dos localidades, diferencia de longitudes.
DELTALAT EQU  $35BC      ;Dos localidades, diferencia de latitudes.
RCTRL    EQU   $10C6      ;Registro de control de casos de orientación.
ANGALFA  EQU   $10C7      ;Una localidad de memoria, ángulo alfa.
ANGGAMA  EQU   $10C8      ;Una localidad de memoria, ángulo gamma.
SH        EQU   $10C9      ;Signo de la diferencia de longitudes de gamma.
SV        EQU   $10CA      ;Signo de la diferencia de latitudes de gamma.
SIGNO    EQU   $10CB      ;Una localidad de memoria, guarda un signo auxiliar.
ANG       EQU   $10CC      ;Guarda magnitud de ángulo alfa o gamma.
SHA       EQU   $10DD      ;Signo de la diferencia de longitudes de alfa.
SVA       EQU   $10DE      ;Signo de la diferencia de latitudes de alfa.

```

## 4.1. SUBROUTINA INSAINI, GUARDA COORDENADAS INSTANTÁNEAS EN INICIALES.

```

INSAINI   ORG    $35BE      ;Dirección de inicio del programa.
          LDX    #$0637
          STX    LATINI    ;Dirección de inicio de latitud inicial.
          LDX    #$0645
          STX    LONINI    ;Dirección de inicio de longitud inicial.
          LDX    #$0718
          STX    DELTALONG ;Dirección de inicio de delta de longitud.
          LDX    #$0727
          STX    DELTALAT  ;Dirección de inicio de delta de latitud
          LDX    #LIMPIA   ;Limpia las localidades de memoria que
          LDY    LATINI    ;utiliza el programa.
          JSR    GDCOOR
          LDX    #LIMPIA
          LDY    LONINI
          JSR    GDCOOR
          LDX    LONINS    ;Carga dir. de long. instantánea en X, (de GPS.asm)
          LDY    LONINI    ;Carga dir. de long. inicial en Y.
          JSR    GDCOOR    ;Guarda coordenada long. instantánea en inicial.
          LDX    LATINS    ;Carga dir. de latitud instantánea en X.
          LDY    LATINI    ;Carga dir. de latitud inicial en Y.
          JSR    GDCOOR    ;Guarda coordenada de lat. instantánea en inicial.
          RTS

```

FIN DE SUBROUTINA INSAINI, REGRESA A MAIN

## 4.2. SUBROUTINA GAMA, CALCULA ANGULO GAMMA ENTRE POSICIÓN INSTANTÁNEA E INICIAL

CALCULA DELTALONG PARA GAMMA, DIFERENCIA DE LONGITUDES

```

GAMA     CLI      ;Habilita interrupciones.
          LDX    #LIMPIA  ;Carga en X dir. de un conjunto de 'espacios'
          LDY    #RESULT  ;Carga en Y dir. de Resultado.
          JSR    GDCOOR   ;Limpia las localidades de RESULT.
          LDX    LONINS   ;Guarda en X dir. de longitud instantánea.
          LDY    LONINI   ;Guarda en Y dir. de longitud inicial $0645.
          JSR    RESTA    ;Calcula diferencia entre LONINS y LONINI.
          LDX    #RESULT  ;Carga en X la dir. donde estará el resultado.
          JSR    FLTASC   ;Convierte resultado en ascii y lo guarda.
          LDAA   RESULT   ;Guarda signo para identificar caso de orientación.
          STAA   SH
          LDX    #RESULT  ;Carga en X dir. inicial en donde está el resultado.
          LDY    DELTALONG ;Carga en Y dir. inicial donde estará la dif. de long,
          JSR    GDCOOR1  ;($0718 a $0725) y guarda resultado desde DELTALONG.
          LDX    DELTALONG ;Carga X con dir. inicial de dif. de longitudes.
          LDY    DELTALONG ;Carga en Y la misma dir. para incrementarla.
          LDAB   #$0E     ;Incrementa Y para tener la dir. final de DELTALONG
          ABY      ;sumando B+Y=733, la dir final.
          JSR    DESP     ;Despliega dif. entre longitud instantánea e inicial.

```

### CALCULA DELTALAT PARA GAMMA, DIFERENCIA DE LATITUDES

```

LDX  LATINS      ;Carga en X dir. inicial de latitud instantánea.
LDY  LATINI      ;Carga en Y dir. inicial de latitud inicial $0637.
JSR  RESTA       ;Calcula dif. entre latitud instantánea e inicial.
LDX  #RESULT     ;Carga en X dir. donde estará resultado.
JSR  FLTASC      ;Convierte el resultado en ascii y lo guarda.
LDAA RESULT      ;Guarda signo para identificar caso de orientación.
STAA SV
LDX  #RESULT     ;Carga en X dir. inicial en donde está el resultado
LDY  DELTALAT    ;Carga en Y dir. inicial donde estará dif. de lat.
JSR  GDCOORD1    ;($0727 a $0734) y guarda en memoria DELTALAT.
LDX  DELTALAT    ;Carga X con dir. inicial de dif. de latitudes $0727.
LDY  DELTALAT    ;Carga en Y la misma dir. para incrementarla.
LDAB #SOC        ;Incrementa Y para tener la dir. final de DELTALAT
ABY          ;sumando B+Y=733, la dir. final.
JSR  DESP        ;Despliega dif. entre latitud instantánea e inicial.

```

### CALCULA LA TANGENTE DE GAMMA, COCIENTE: DELTALONG / DELTALAT

```

LDX  DELTALAT    ;Carga en X dir. de diferencia de latitudes $727.
INX
LDY  DELTALONG   ;Carga en Y dir. de diferencia de longitudes $718.
INY
JSR  DIVISION    ;Calcula el cociente entre DELTALAT y DELTALONG
JSR  CHKTAN      ;Verifica signo del valor y obtiene el ángulo.
JSR  DRDC        ;Despliega registro de control
LDX  #RESULT     ;Transforma el resultado,
JSR  FLTASC      ;de flotante a ascii.
LDX  #RESULT     ;Despliega el resultado.
LDY  #$313E
JSR  DESP
LDAB ANG         ;Carga en B el valor dado por CHKTAN
STAB ANGGAMA     ;y guarda ese ángulo en la variable ANGGAMA.
RTS              ;Termina subrutina.

```

### 4.3. SUBROUTINA LLENAG, LLENA EL REGISTRO DE CONTROL PARA EL CASO DE GAMMA

```

LLENAG  CLR  RCTRL    ;Limpia la variable del registro de control.
        LDX  #$1000   ;Carga X para usar modo indexado.
        LDAA #2D      ;Carga el código hexadecimal de un (-).
        CMPA SH       ;Compara con el signo de la diferencia de longitudes.
        BEQ  CSVG     ;Si signo es negativo salta a comparar el otro signo.
        CMPA SV       ;Si NO es negativo compara signo de la dif. De lat.
        BEQ  RUTD     ;Si la comparación es positiva, escribirá CASO D.
        BSET %C6,X,$41 ;(++ RUTINA A, 41h, 65d, A Código de CASO A.
        RTS
CSVG    CMPA SV       ;Signo de long. Negativo, compara signo de latitud.
        BEQ  RUTC     ;Si signo de lat. Es negativo, escribirá CASO C.
        BSET %C6,X,$51 ;(-+ RUTINA B, 51h, 81d, Q Código de CASO B.
        RTS
RUTD    BSET %C6,X,$71 ;(+--) RUTINA D, 71h, 113d, q Código de CASO D.
        RTS
RUTC    BSET %C6,X,$61 ;(-- RUTINA C, 61h, 97d, a Código de CASO C.
        RTS

```

FIN DE SUBROUTINA LLENAG

4.4. SUBROUTINA BETA,  
OBTIENE LA MAGNITUD Y SENTIDO DE LA CORRECCIÓN DE TRAYECTORIA DEL MÓVIL

|       |      |         |  |
|-------|------|---------|--|
| BETA  | LDAA | RCTRL   | ;Carga el registro de control en A.                |
|       | CMPA | #\$57   | ;Compara con 57h, código de Caso B3.               |
|       | BEQ  | B3      | ;Si el registro es 57h, salta a CASO B3.           |
|       | CMPA | #\$73   | ;Si el registro de control es 73h                  |
|       | BEQ  | B3      | ;también es CASO B3.                               |
|       | CMPA | #\$4D   | ;Compara con 4Dh, código de Caso A4.               |
|       | BEQ  | A4      | ;Si el registro es 4Dh, salta a CASO A4.           |
|       | CMPA | #\$65   | ;Si el registro de control es 65h                  |
|       | BEQ  | A4      | ;también es CASO A4                                |
|       | CMPA | #\$53   | ;53h es el código de CASO B1.                      |
|       | BEQ  | B1      |  |
|       | CMPA | #\$77   | ;77h también indica CASO B1.                       |
|       | BEQ  | B1      |  |
|       | CMPA | #\$45   | ;45h código de CASO A2.                            |
|       | BEQ  | A2      |  |
|       | CMPA | #\$6D   | ;6Dh también es código de CASO A2.                 |
|       | BEQ  | A2      |  |
|       | BRA  | SIGUE   | ;En resto de casos se compara magnitud de ángulos. |
| A4    | CLR  | DIR     | ;Para A4 limpia variable DIR, mov. a la izquierda. |
|       | BRA  | DE      | ;Calcula Beta= Alfa + Gamma.                       |
| B3    | LDAA | #\$DE   | ;Caso B3 carga DEh en DIR, mov. a la derecha.      |
|       | STAA | DIR     |  |
| DE    | LDAA | ANGALFA | ;Calcula Beta= Alfa + Gamma.                       |
|       | ADDA | ANGGAMA |  |
|       | STAA | GRADOS  | ;Guarda resultado en GRADOS                        |
|       | RTS  |         | ;Termina subrutina, casos A4 o B3.                 |
| B1    | CLR  | DIR     | ;Para B1, limpia DIR, mov. a la izquierda.         |
|       | BRA  | NI      | ;Calcula Beta= 180 - (alfa + gamma).               |
| A2    | LDAA | #\$DE   | ;Para caso A2, movimiento a la derecha.            |
|       | STAA | DIR     |  |
| NI    | LDAA | #\$B4   | ;Carga un 180d = B4h en A.                         |
|       | LDAB | ANGALFA | ;Carga alfa en B.                                  |
|       | ADDB | ANGGAMA | ;Suma Alfa + Gamma, resultado en B.                |
|       | SBA  |         | ;Resta A - B = 180 - (alfa + gamma).               |
|       | STAA | GRADOS  | ;Guarda el resultado en GRADOS.                    |
|       | RTS  |         | ;Termina subrutina para casos B1 o A2.             |
| SIGUE | LDAA | ANGGAMA | ;Carga ángulo alfa en A.                           |
|       | LDAB | ANGALFA | ;Carga ángulo gamma en B.                          |
|       | CBA  |         | ;Compara Alfa con Gamma.                           |
|       | BMI  | AMG     | ;Si alfa es menor que Gamma, salta a AMG.          |
|       | LDAA | RCTRL   | ;Carga registro de control en A.                   |
|       | CMPA | #\$43   | ;Compara con 43h, código de caso A1.               |
|       | BEQ  | A1      | ;Si el registro es 43h salta a CASO A1.            |
|       | CMPA | #\$67   | ;Compara con 67h, también es código de A1.         |
|       | BEQ  | A1      | ;Si el registro es 67h, salta a CASO A1.           |
|       | CMPA | #\$55   | ;Si el registro es 55h, salta a CASO B2.           |
|       | BEQ  | B2      |  |
|       | CMPA | #\$7D   | ;Si el registro es 7Dh, también salta a CASO B2.   |
|       | BEQ  | B2      |  |
|       | CMPA | #\$5D   | ;Si el registro es 5D, salta a CASO B4.            |
|       | BEQ  | B4      |  |
|       | CMPA | #\$75   | ;Si el registro es 75h, también salta a CASO B4    |
|       | BEQ  | B4      |  |

```

      CMPA  #$47      ;Si el registro es 47h, salta a CASO A3.
      BEQ   A3
      CMPA  #$63      ;Si el registro es 63h, también salta a CASO A3.
      BEQ   A3
      LDAA  #$64      ;Si el registro no corresponde a ningún caso,
      STAA  RCTRL     ;carga el código de una arroba, lo guarda en el
      JSR   DRDC      ;registro y lo despliega como mensaje de error.
      RTS                ;Termina subrutina.

AMG   LDAA  RCTRL     ;Cuando Alfa > Gamma
      CMPA  #$43      ;Compara registro de control.
      BEQ   A1X       ;Si registro es 43h, CASO A1X (alfa>gamma).
      CMPA  #$67      ;Si registro es 67h, también es CASO A1X.
      BEQ   A1X
      CMPA  #$55      ;Si registro es 55h, CASO B2X.
      BEQ   B2X
      CMPA  #$7D      ;Si registro es 7Dh, también es CASO B2X.
      BEQ   B2X
      CMPA  #$5D      ;Si registro es 5Dh, CASO B4X
      BEQ   B4X
      CMPA  #$75      ;Si registro es 75h, también es CASO B4X.
      BEQ   B4X
      CMPA  #$47      ;Si registro es 47h, CASO A3X.
      BEQ   A3X
      CMPA  #$63      ;Si registro es 63h, también es CASO A3X.
      BEQ   A3X
      LDAA  #$64      ;Si no se encontró ningún caso, carga
      STAA  RCTRL     ;mensaje de error en registro de control y
      JSR   DRDC      ;se despliega en pantalla.
      RTS                ;Termina subrutina, regresa a MAIN.

A1    CLR   DIR       ;CASO A1, Gamma >Alfa movimiento a la Izquierda.
      BRA  NIZ
B2    LDAA  #$DE
      STAA  DIR       ;CASO B2, Gamma > Alfa, movimiento a la Derecha.
NIZ   LDAA  ANGGAMA    ;Carga Alfa en A.
      SUBA  ANGALFA    ;Resta: Gamma - Alfa.
      STAA  GRADOS     ;Guarda Beta en GRADOS.
      RTS                ;Regresa a MAIN.

B4    CLR   DIR       ;CASO B4, Gamma > Alfa, movimiento a la Izquierda.
      BRA  SE
A3    LDAA  #$DE
      STAA  DIR       ;CASO A3, Gamma > Alfa, movimiento a la derecha.
SE    LDAA  #$5A
      STAA  GRADOS     ;Carga 5Ah = 90d en A.
      RTS                ;Guarda 90d en GRADOS (Beta = 90)
      RTS                ;Termina subrutina, regresa a MAIN.

B2X   CLR   DIR       ;CASO B2X, Alfa > Gamma movimiento a la izquierda.
      BRA  NIZX
A1X   LDAA  #$DE
      STAA  DIR       ;CASO A1X, Alfa > Gamma movimiento a la derecha.
NIZX  LDAA  ANGALFA    ;Carga Alfa en A.
      SUBA  ANGGAMA    ;Resta: Beta = Alfa - Gamma
      STAA  GRADOS     ;Guarda el resultado en Grados y
      RTS                ;Termina subrutina, regresa a MAIN.

A3X   CLR   DIR       ;CASO A3X, Alfa > Gamma movimiento a la izquierda.
      BRA  FDCX
B4X   LDAA  #$DE
      STAA  DIR       ;CASO B4X, Alfa > Gamma movimiento a la derecha.

```

```

FDCX      LDAA  #55A      ;Carga 5Ah = 90d en A.
          STAA  GRADOS   ;Guarda en GRADOS (Beta = 90).
          RTS           ;Termina subrutina, regresa a MAIN.

```

#### CONSTANTES

```

          ORG  $386A      ;Esta constante es 0.0000
UNO1:    FDB  $302E,$3030 ;Es una constante auxiliar para transformar
          FDB  $3030,$2020 ;ascii en flotante.
          END

```

#### 5. PALFA.ASM

Este programa realiza las operaciones para obtener el ángulo ALFA

#### 5.1. SUBROUTINA ALFA CALCULA ÁNGULO ALFA ENTRE POSICIÓN INSTANTÁNEA Y FINAL

##### CALCULA DELTALONG, DIFERENCIA DE LONGITUDES PARA ALFA

```

ALFA      CLI           ;Habilita interrupciones.
          LDX  #LIMPIA  ;Carga dir. de un conjunto de 'espacios'.
          LDY  #RESULT  ;Carga dir. de resultado.
          JSR  GDCOORD  ;Va a limpiar las dir. del RESULT.
          LDX  LONFIN   ;Carga #5060D dir. de longitud final.
          LDY  LONINS   ;Carga #50629 (0628) dir. de longitud instantánea.
          JSR  RESTA    ;Calcula la diferencia entre LONFIN y LONINS.
          LDX  #RESULT  ;Carga X con la dir. del resultado en flotante.
          JSR  FLTASC   ;Convierte el resultado en ascii.
          LDAA RESULT   ;Guarda signo para identificar caso de orientación
          STAA SHA
          LDX  #RESULT  ;Carga en X dir. del resultado en ascii.
          LDY  DELTALONG ;Carga en Y dir. donde estar la dif. de longitudes,
          JSR  GDCOORD  ;($0718 a $0725) y guarda en memoria DELTALONG.
          LDX  DELTALONG ;Carga en X la dir. inicial de la dif. de longs.
          LDY  DELTALONG ;Carga en Y la misma dir. para incrementarla.
          LDAB #50D     ;Incrementa Y para tener la dir. final de DELTALONG
          ABY          ;sumando B a Y=$0725h, la dir final.
          JSR  DESP     ;Despliega diferencia de longitudes.

```

##### CALCULA DELTALAT, DIFERENCIA DE LATITUDES PARA ALFA

```

          LDX  LATFIN   ;Carga en X la dir. inicial de latitud final $0600.
          LDY  LATINS   ;Carga en Y la dir. inicial de lat. instantánea $0619.
          JSR  RESTA    ;Calcula diferencia entre LATFIN y LATINS.
          LDX  #RESULT  ;Carga X con dir. inicial del resultado en flotante.
          JSR  FLTASC   ;Convierte el resultado en ascii.
          LDAA RESULT   ;Guarda signo para identificar caso de orientación.
          STAA SVA
          LDX  #RESULT  ;Carga en X la dir. del resultado en ascii.
          LDY  DELTALAT ;Carga en Y la dir. inicial para la dif. de latitudes
          JSR  GDCOORD  ;(de $0727 a $0734) y guarda en memoria para DELTALAT.
          LDX  DELTALAT ;Carga en X la dir. inicio de la dif. de latitudes.
          LDY  DELTALAT ;Carga en Y la misma dir. para incrementarla.
          LDAB #50C     ;Incrementa Y para tener la dir. final de DELTALAT
          ABY          ;sumando B a Y=$0733.
          JSR  DESP     ;Despliega diferencia de latitudes.

```

**CALCULA TANGENTE DE ALFA, COCIENTE: DELTALAT/DELTALONG**

```

LDX  DELTALAT ;Carga en X dir. de diferencia de latitudes ($0727).
INX                      ;Incrementa en uno para evitar el signo.
LDY  DELTALONG ;Carga en Y dir. de diferencia de longitudes($0718).
INY                      ;Incrementa en uno para evitar el signo.
JSR  DIVISION  ;Calcula cociente entre DELTALAT y DELTALONG.

```

**VERIFICA EL SIGNO DEL VALOR OBTENIDO Y OBTIENE EL ÁNGULO**

```

JSR  CHKTAN   ;Verifica signo del valor obtenido y obtiene el ángulo.
JSR  DRDC     ;Despliega el registro de control para monitoreo.
LDX  #RESULT  ;Convierte el resultado, en FPAC1 a código ASCII.
JSR  FLTASC   ;Despliega el resultado, de FPAC1.
LDX  #RESULT  ;Despliega el resultado, de FPAC1.
LDY  #$313D   ;Despliega el resultado, de FPAC1.
JSR  DESP     ;Despliega el resultado, de FPAC1.
LDAB ANG      ;Carga el valor del ángulo en B.
STAB ANGALFA  ;El contenido de B lo guarda en variable ANGALFA
RTS          ;Termina subrutina.

```

FIN DE SUBROUTINA ALFA, REGRESA A MAIN

**5.2. SUBROUTINA LLENAA, LLENA EL REGISTRO DE CONTROL CON EL CASO DE ALFA**

```

LLENAA  LDX    #$1000      ;Carga 1000h en X para direccionamiento indexado.
        LDAA   #$2D       ;Carga en A el valor 2Dh = signo negativo.
        CMPA  SHA        ;Compara A con el signo de longitud de alfa.
        BEQ   CSVA       ;Si signo es (-) comparará el signo de latitud.
        CMPA  SVA        ;Compara el signo de latitud.
        BEQ   RUT4       ;Signo latitud=(-), RUTINA 4, si no, escribe:
        BSET  $C6,X,$02   ;(++ ) RUTINA 1, 02h, 02d, HappyFace en registro.
        RTS          ;Termina subrutina, RUTINA 1.
CSVA    CMPA  SVA        ;Compara el signo de latitud de alfa.
        BEQ   RUT3       ;Si es negativo, RUTINA 3, si no, escribe:
        BSET  $C6,X,$04   ;(-+ ) RUTINA 2, 04h, 04d, Diamond
        RTS          ;Termina subrutina, RUTINA 2.
RUT4    BSET  $C6,X,$0C   ;(+ - ) RUTINA 4, 0Ch, 12d, Female
        RTS          ;Termina subrutina, RUTINA 4.
RUT3    BSET  $C6,X,$06   ;(-- ) RUTINA 3, 06h, 06d, Espadas
        RTS          ;Termina subrutina, RUTINA 3.

```

FIN DE SUBROUTINA LLENAA

**5.3. SUBROUTINA CHKTAN, OBTIENE EL VALOR EN FORMATO FLOTANTE DEL VALOR TANGENTE EN LA TABLA EN ASCII**

```

CHKTAN  LDAA  RESULT     ;Carga en A byte de signo de la división.
        LDAB  #$2D       ;Cuando es negativo el byte de signo es 2D (-).
        CBA                      ;Compara A con B, se restan pero no cambian.
        BEQ  NEG         ;Si la bandera Z (cero) se activa, fue negativo.
        CLR  SIGNO       ;Se limpia la variable SIGNO.
        JMP  CTT         ;Salta a Compara Tabla Tangente.
NEG     LDAA  #$30       ;Al ser (-), entonces lo pasa a positivo
        STAA RESULT     ;colocando 00h en el byte de signo, haciéndolo (+)
        LDAA  #$2D       ;colocando 00h en el byte de signo, haciéndolo (+)
        STAA  SIGNO     ;Guarda el (-) en la variable SIGNO.

```

```

CTT      ;LDX  #RESULT  ;Despliega resultado en ASCII
        ;LDY  #RESULT
        ;LDAB #SOC
        ;ABY
        ;JSR  DESP
        LDX  #RESULT  ;Carga la tangente en X.
        LDY  #UNO1    ;Carga la constante 0.0000 en Y
        JSR  SUMA     ;Suma y así transforma la variable a flotante.
        LDX  #$2AFB   ;Carga en X dir. cercana a tabla tangentes flotantes.
COMPARA  INX          ;Incrementa X para tener el primer valor de la tabla
        INX          ;o, dentro del ciclo, para tener el siguiente valor
        INX          ;de la tabla de tangentes.
        INX
        INX
        JSR  GETFPAC2 ;Guarda el flot. que está en la dir de X en FPACC2.
        JSR  FLTCMP   ;Compara FPACC1 con FPACC2 (FPACC1-FPACC2).
        BHS  COMPARA  ;Si el resultado es mayor que cero, va a COMPARA
SENDGRAD STX  ANG      ;Si el resultado es igual o menor que cero,
        LDX  #ANG     ;guarda el contenido de X (dir. de tangente) en ANG y
        LDY  #ANG     ;Despliega el valor de la dirección.
        INY
        INY
        JSR  DESP
        LDX  $3664    ;Carga en X la variable ANG (está en dir. 3664h).
        XGDX         ;Intercambia X con D.
        CPD  #$2B5A   ;Compara D con 2B5Ah, que es el límite de tabla.
        BHI  LIM      ;Si D es mayor que el límite, salta a LIM.
        CLRA         ;Si D NO es mayor, limpia parte alta de D (A) para
        ;tener la dif. entre dir. de flot. y origen de tabla.
        LDX  #$0005   ;Carga 5 en X para obtener el intervalo en que está
        IDIV         ;el valor de la tangente que se comparó, cociente en X
        XGDX         ;el cociente lo pasa a D (acumulador B).
        LDAA #03      ;Carga 3, incremento en grados en A para obtener los
        MUL          ;grados = A*B y el valor queda en la parte baja de D.
        STAB $3664    ;Guarda el valor del ángulo en var ANG (dir. 3664h)
        LDX  #ANG     ;Se despliega ANG, primera vez es Alfa, segunda vez es
        LDY  #ANG     ;Gamma.
        INY
        INY
        JSR  DESP
        LDAB $3664    ;Compara valor del ángulo tanto en alfa
        CMPB #$57     ;como en gamma con el valor límite que es 87d (57h)
        BHS  LIM      ;Si el valor es mayor o igual, va a LIM.
        RTS          ;Si no es mayor o igual a 87d, termina subrutina.
LIM      LDAB #$57     ;Si pasa el límite de dirección o de valor de ángulo,
        STAB ANG      ;se guarda un 90d en ANG, valor máximo permitido
        LDX  #ANG
        LDY  #ANG     ;Despliega el ángulo, alfa o gamma según corresponda.
        INY
        INY
        JSR  DESP
        RTS          ;Termina subrutina.

```

FIN DE SUBROUTINA CHKTAN, REGRESA A ALFA O A GAMA SEGÚN SEA EL CASO



## 6. MOTOR.ASM

Contiene las subrutinas que controlan a los motores de dirección y propulsión

## DECLARACIÓN DE VARIABLES

|         |     |        |  |
|---------|-----|--------|--|
| ENTERO  | EQU | \$3872 | ;Una localidad, la utiliza la subrutina NUMERO.        |
| TALTO   | EQU | \$3874 | ;TALTO 16 bits, dos localidades.                       |
| TBAJO   | EQU | \$3876 | ;TBAJO 16 bits, dos localidades.                       |
| GRADOS  | EQU | \$3878 | ;Una localidad, la utiliza la subrutina NUMERO.        |
| DIR     | EQU | \$387A | ;Para indicar mov. derecha o izquierda.                |
| EDO     | EQU | \$387C | ;Guarda la posición de PORTB al leerlo.                |
| ESTADOS | EQU | \$387E | ;Numero de cambios para mover motor de dirección.      |
| HIGH    | EQU | \$3880 | ;Tiempo en estado alto del Ciclo de Trabajo (CT)       |
| ENTERO  | EQU | \$3882 | ;Variable para encontrar intervalo del ángulo de giro. |
| DIREC   | EQU | \$3884 | ;Sentido de giro del motor de dirección.               |

## 6.1. SUBROUTINA MOTORP, ENCIENDE MOTOR DE PROPULSIÓN Y ACTUALIZA CICLO DE TRABAJO

|        |        |             |   |
|--------|--------|-------------|---|
| ORG    | \$388A |             | ;Posicion inicial del programa en RAM.              |
| MOTORP | LDX    | #\$1000     | ;Carga dir. base de Indice X.                       |
|        | BSET   | \$22,X,\$20 | ;Enciende bit OC3I en TMSK1, habilita interrupción. |
|        | BCLR   | \$23,X,\$DF | ;Borra bandera OC3F en TFLG1.                       |
|        | BSET   | \$20,X,\$10 | ;Enciende bit OL3 en TCTL1 para modo Toggle.        |
|        | CLR    | \$3880      | ;Limpia la variable HIGH.                           |
|        | LDAA   | #\$03       | ;Configura PORTC con 2 pines de salida (PC0,PC1) y  |
|        | STAA   | DDRC        | ;6 entrada (PC2,PC3,PC4,PC5,PC6,PC7) para sensores. |
|        | LDX    | #\$0320     | ;0320h = 800d.                                      |
|        | STX    | TALTO       | ;Carga 800 en TALTO, 80% del ciclo de trabajo.      |
|        | LDX    | #\$00C8     | ;00c8h = 200d.                                      |
|        | STX    | TBAJO       | ;Carga 200 en TBAJO, 20% del ciclo de trabajo.      |
|        | CLI    |             | ;Habilita interrupción.                             |
|        | RTS    |             | ;Regresa de subrutina.                              |

FIN DE SUBROUTINA MOTORP, REGRESA A MAIN

6.2. ISR DE MOTOR DE PROPULSIÓN,  
ACTUALIZA CICLO DE TRABAJO DEL MOTOR DE PROPULSIÓN

|      |        |   |
|------|--------|---|
| ORG  | \$38C0 | ;Dirección de interrupción de TOC3.             |
| LDAA | #\$10  |   |
| STAA | TCTL1  | ;Configura modo Toggle para OC3.                |
| LDAA | \$3880 | ;Carga la variable HIGH.                        |
| BNE  | LOW    | ;Si HIGH != 0, va a LOW, actualizar ciclo bajo. |

## ACTUALIZA CICLO DE TRABAJO ALTO

|      |        |  |
|------|--------|--|
| LDD  | TCNT   | ;Carga TCNT (Timer Counter Register) en D.           |
| ADDD | TALTO  | ;Suma TALTO + TCNT y                                 |
| STD  | TOC3   | ;guarda el valor en registro de comp. TOC3.          |
| LDAA | #\$20  |  |
| STAA | TFLG1  | ;Limpia bandera OC3F en TFLG1.                       |
| INC  | \$3880 | ;Incrementa HIGH.                                    |
| RTI  |        | ;Fin de interrupción luego de actualizar ciclo alto. |

## LOW, ACTUALIZA CICLO DE TRABAJO BAJO

```

LOW   LDD      TCNT      ;Carga TCNT (Timer Counter Register) en D.
      ADDD     TBAJO     ;Suma TBAJO + TCNT y
      STD      TOC3     ;guarda el valor en registro de comp. TOC3.
      LDAA    #$20
      STAA    TFLG1    ;Limpia bandera OC3 en TFLG1.
      CLR     $3880    ;Limpia variable HIGH, indica ciclo por actualizar.
      RTI

```

## FIN DE LA ISR PARA MOTOR DE PROPULSIÓN

## 6.3. SUBROUTINA PASOS, CONTROLA EL MOVIMIENTO DEL MOTOR DE DIRECCIÓN

```

PASOS LDY      #$1000    ;Carga la dir. base de Indice X.
      BCLR     $03,Y,$01 ;Limpia BIT0 de PORTC, APAGA SENSORES.
      LDAA    DIR       ;Carga en A la variable DIR.
      BEQ     IZQ       ;Si DIR = 0 va a mover motor a la izquierda.

```

## MUEVE MOTOR A LA DERECHA

```

DER   LDAA    PORTB     ;Lee el estado del motor de dirección.
      LDX     #$10B8    ;#$0420 Carga en X dir. inicial de mov.
      CMPA   #$01       ;Compara con edo. 01.
      BEQ    MDER      ;Si es igual va a moverse a la derecha.
      INX
      CMPA   #$02       ;Incrementa X para tener el siguiente estado de tabla.
      BEQ    MDER      ;Compara con edo. 02.
      INX
      CMPA   #$04       ;Si es igual va a moverse a la derecha.
      BEQ    MDER      ;Incrementa X para tener el sig. edo. de tabla.
      INX
      CMPA   #$08       ;Compara con edo. 04.
      BEQ    MDER      ;Si es igual va a moverse a la derecha.
      INX
      CMPA   #$08       ;Incrementa X para tener el sig. edo. de tabla.
      BEQ    MDER      ;Compara con edo. 08.
      LDX     #$10B8    ;Si NO cumple ninguna condición, carga en X edo.01.
MDER  LDAB    ESTADOS   ;Carga estados en contador B.
MD    DEX
      LDAA    0,X       ;Decrementa X para tener el estado anterior de tabla.
      STAA   PORTB     ;Carga en A el contenido de la dir. X.
      LDY    $0FF8     ;Pone en Puerto B el estado correspondiente a esa dir.
DYY   DEY
      BNE    DYY
      DECB   ;Tiempo de polarización de motor de dirección
      BNE    MD        ;Decrementa B que es el contador de estados.
      STX   DIR        ;Mientras B!=0, repite la operación, mueve motor.
      JSR   ESPERA     ;Mitad del movimiento, guarda X en DIR.
      LDY   DIR        ;Tiempo para corrección de trayectoria.
      LDAB  ESTADOS    ;Carga DIR en Y (el valor de estados).
MIZD  LDAB    ESTADOS   ;Carga estados en B, contador.
MIZD1 INY
      LDAA  0,Y        ;Incrementa Y, el siguiente estado.
      STAA  PORTB     ;Carga el valor en A.
      LDX  $0FF8     ;Pone el valor de A en el Puerto B, motor de dirección.
      DEX
DYY1  DEX
      BNE   DYY1
      DECB ;Tiempo de polarización para motor de dirección.
      BNE  MIZD1     ;Decrementa B, contador de estados.

```

```

CLI                ;Terminó movimiento, habilita interrupciones.
CLR                DIR        ;Limpia variable DIR.
LDY                #$1000    ;ENCIENDE SENSORES.
BSET               $03,Y,$01
RTS                ;Termina subrutina, el movimiento fue a la derecha.

```

### MUEVE MOTOR A LA IZQUIERDA

```

IZQ                LDAA        PORTB    ;Lee el estado del motor de dirección.
                   LDX         #$10A8  ;Carga en X dir. inicial de mov.
                   CMPA        #$01    ;Compara con edo. 01.
                   BEQ         MIZQ    ;Si es igual va a moverse a la izquierda.
                   INX         ;Incrementa X para tener el siguiente estado de tabla.
                   CMPA        #$02    ;Compara con edo. 02.
                   BEQ         MIZQ    ;Si es igual va a moverse a la izquierda.
                   INX         ;Incrementa X para tener el sig. edo. de tabla.
                   CMPA        #$04    ;Compara con edo. 04.
                   BEQ         MIZQ    ;Si es igual va a moverse a la izquierda.
                   INX         ;Incrementa X para tener el sig. edo. de tabla.
                   CMPA        #$08    ;Compara con edo. 08
                   BEQ         MIZQ    ;Si es igual va a moverse a la izquierda.
                   LDX         #$10A7  ;Si NO cumple ninguna condición, carga en X el edo. 01.
MIZO                LDAB        ESTADOS ;Carga estados en contador B.
MI                 INX         ;Incrementa X para tener el sig. estado de la tabla.
                   LDAA        0,X     ;Carga en A el contenido de la dir. X.
                   STAA        PORTB  ;Pone en Puerto B el estado correspondiente a esa dir.
                   LDY         #$0FF8  ;Tiempo para polarizar el motor de dirección.
DY                 DEY
                   BNE         DY
                   DECB        ;Decrementa B que es el contador de estados.
                   BNE         MI      ;Mientras B;=0, repite la operación, mueve motor.
                   STX         DIREC   ;Mitad del movimiento, guarda X en DIREC y
                   JSR         ESPERA  ;espera para corrección de trayectoria.
                   LDY         DIREC   ;Carga DIREC en Y.
MDI                LDAB        ESTADOS ;Carga Estados en B.
MDI1               DEY         ;Decrementa en uno Y (estado anterior).
                   LDAA        0,Y     ;Carga el valor de Y en A.
                   STAA        PORTB  ;Coloca el valor en Puerto B, motor de dirección.
                   LDX         #$0FF8  ;Tiempo para polarizar el motor de dirección
DY1                DEX
                   BNE         DY1
                   DECB        ;Decrementa B, contador de estados.
                   BNE         MDI1    ;Mientras B !=0, repite la operación, mueve motor.
                   CLI         ;Terminó movimiento, habilita interrupciones.
                   CLR         DIR     ;Limpia variable DIR.
                   LDY         #$1000  ;ENCIENDE SENSORES después de completar movimiento.
                   BSET        $03,Y,$01
                   RTS                ;Termina subrutina, el movimiento fue a la izquierda.

```

FIN SUBROUTINA PASOS

**6.4. SUBROUTINA NUMERO,  
CALCULA EL NÚMERO DE ESTADOS QUE SE USA PARA EL GIRO DEL MOTOR DE DIRECCIÓN**

```

NUMERO  LDX    #GRADOS    ;Despliega Grados.
        LDY    #GRADOS
        INY
        JSR    DESP
        LDD    $3870     ;Carga GRADOS (dir. 3870h) en D.
        LDX    #$0D      ;Carga en X 13, escala de grados por paso.
        IDIV   ;Divide D entre X.
        STX    ENTERO    ;Guarda el cociente en entero.
        LDAA   ENTERO    ;Carga el entero en A.
        ADDA   #$01      ;Suma 1 al entero que está en A.
        LDAB   ENTERO    ;Carga el entero en B.
        ABA    ;Suma A con B (entero + entero + 1).
        STAA   ESTADOS   ;Obtiene número de estados para girar motor.
        LDAA   #$0D      ;Carga en A 0Dh = 13d.
        CMPA   ESTADOS   ;Compara con el valor de ESTADOS.
        BMI    MAYOR13   ;Si estados es mayor a 13d, va a MAYOR13
SAL     LDX    #ESTADOS  ;Despliega Estados.
        LDY    #ESTADOS
        INY
        JSR    DESP
        RTS             ;Termina subrutina.
MAYOR13 STAA   ESTADOS   ;Guarda 13d en estados.
        BRA    SAL      ;Va a desplegar estados y terminar subrutina.

```

**FIN DE SUBROUTINA NUMERO**

**6.5. SUBROUTINA ESPERA, PRODUCE TIEMPO PARA CORRECCIÓN DE TRAYECTORIA**

```

ESPERA  LDX    #$000F
        LDY    #$0012
T1      DEX
        BNE   T1
        DEY
        BNE   T1
        RTS

```

**FIN DE SUBROUTINA ESPERA**

**VECTOR DE INTERRUPCIÓN DE MOTOR.ASM**

```

ORG    $00D9    ;Vector de Interrupción del motor de propulsión
JMP    $38C0

```

## 7. SENSOR.ASM

Programa que contiene las subrutinas que controlan el sistema de sensores

## 7.1 SUBROUTINA SENSOR, HABILITA EL SISTEMA DE SENSORES

```

SENSOR   ORG     $3C30           ;Dirección de inicio en memoria.
         LDAA    #$03           ;Se configuran 2 pines de salida (PC0, PC1) y 6 de
         STAA    DDRC          ;entrada (PC2, PC3, PC4, PC5, PC6, PC7) en PORTC.
         LDX     #$1000
         BSET    $03,X,$01
         CLI     ;Habilita interrupciones.
         RTS     ;Termina subrutina.

```

## 7.2 ISR DE SENSORES, DETERMINA QUÉ SENSOR SE HA ACTIVADO

```

SENS     ORG     $3C50           ;Dirección de inicio de la ISR.
         LDAA    PORTC         ;Lee el contenido de PORTC.
         LDX     #$1000
         BCLR    $03,X,$01      ;APAGA SENSORES.
         CMPA    #$21          ;Compara con 20h que es código del sensor CENTRAL.
         BEQ     S2            ;Salta a subrutina S2.
         CMPA    #$31          ;Compara con 30h, código sensor DERECHO+CENTRAL.
         BEQ     S3            ;Salta a subrutina S3.
         CMPA    #$61          ;Compara con 60h, código sensor IZQUIERDO+CENTRAL.
         BEQ     S6            ;Salta a subrutina S6.
         CMPA    #$71          ;Compara con 70h, todos los sensores activados.
         BEQ     S7            ;Salta a subrutina S7.
         LDX     #$1000        ;Carga 1000h en X para direccionamiento indexado.
         BSET    $03,X,$01      ;Enciende en la dir. 1003h el bit0.
         CLI     ;Habilita interrupciones.
         LDAA    #$20
         STAA    TFLG1         ;Limpia bandera
         RTI     ;Termina interrupción.

```

Para Sensor Central: Motor de dir. gira 15 grados a la izquierda

```

S2       CLR     DIR           ;Limpia variable DIR (mov. a la izquierda)
         LDAA    #$03
         STAA    ESTADOS       ;Carga 03d en la variable ESTADOS (15 grados)
         JSR     PASOS         ;Va a subrutina PASOS, mueve motor de dir.
         LDY     #$1000        ;Se usa X en modo indexado
         BSET    $03,Y,$01      ;Enciende sensores
         LDAA    #$20
         STAA    TFLG1         ;Limpia bandera
         CLI     ;Habilita interrupciones y
         RTI     ;regresa a esperar un nuevo edo. de sensores.

```

Para Sensor Derecho + Central: Motor de dir. gira 60 grados a la izquierda.

```

S3       CLR     DIR           ;Limpia variable DIR (mov. a la izquierda).
         LDAA    #$08
         STAA    ESTADOS       ;Carga 08d en ESTADOS, (60 grados).
         JSR     PASOS         ;Va a subrutina PASOS, mueve motor de dir.
         LDX     #$1000        ;X se usa en modo indexado
         BSET    $03,X,$01      ;Enciende sensores y
         CLI     ;habilita interrupciones.
         LDAA    #$20
         STAA    TFLG1         ;Borra bandera y
         RTI     ;termina interrupción.

```

Para Sensor Izquierdo + Central: Motor de dir. gira 60 grados a la derecha.

```
S6   LDAA   #$DE           ;Carga DEh en DIR (mov. a la derecha).
      STAA  DIR
      LDAA  #$08           ;Carga 08h en ESTADOS (60 grados).
      STAA  ESTADOS
      JSR   PASOS          ;Va a subrutina PASOS, mueve motor de dir.
      LDX   #$1000        ;X se usa en modo indexado.
      BSET  $03,X,$01     ;Enciende sensores y
      CLI   ;Habilita interrupciones.
      CLR   DIR           ;Limpia variable DIR.
      LDAA  #$20
      STAA  TFLG1         ;Borra bandera y
      RTI   ;termina interrupción.
```

Para todos sensores activados: Disminuye vel. y gira 90 grados a la derecha.

```
S7   LDAA   #$DE           ;Guarda DEh en DIR (mov. a la derecha).
      STAA  DIR
      LDAA  #$0D           ;Guarda 13d en estados (90 grados).
      STAA  ESTADOS
      JSR   PASOS          ;Va a subrutina PASOS, mueve motor de dirección.
      LDX   #$1000        ;X se usa en modo indexado.
      BSET  $03,X,$01     ;Enciende sensores y
      CLI   ;habilita interrupciones.
      CLR   DIR           ;Limpia la variable DIR.
      LDAA  #$20           ;Borra bandera.
      STAA  TFLG1
      RTI   ;Termina la interrupción.
```

#### VECTOR DE INTERRUPCIÓN DE SENSOR.ASM

```
ORG   $00EE           ;Vector de interrupción del sistema de sensores.
JMP   $3C50
END
```

#### 8. COMPARA.ASM

Este archivo realiza la comparación entre el límite de acercamiento y la posición en la que se encuentra el vehículo

##### 8.1. SUBROUTINA LLEGA, VERIFICA CONDICIÓN DE LÍMITE DE ACERCAMIENTO

```
LLEGA  ORG   $3F40       ;Dirección de inicio del programa en memoria.
        LDX   LONINS     ;Calcula la diferencia entre la longitud inicial y la
        LDY   LONFIN     ;final, la cual se comparará con el valor límite de
        JSR   RESTA      ;acercamiento a la coordenada final. En caso de que
        CLR   $0004      ;DELTALONG sea negativa se borra el signo para
        LDX   #RESULT    ;hacer la comparación con valores absolutos.
        JSR   FLTASC
        LDX   #RESULT    ;Al valor de DELTALONG le resta el valor del límite
        LDY   #LIMITE    ;y si el resultado es negativo, significa que el límite
        JSR   RESTA      ;es mayor que DELTALONG, por lo que se considera que el
        LDX   #RESULT    ;móvil llegó a su destino.
        JSR   FLTASC
        LDAA  $0004      ;Para verificar el signo de la resta lee el bit de signo
        BNE   LATCOMP    ;de FPACCl. Si no es positivo (00h), va a LATCOMP.
        BRA   EXIT       ;Si es positivo, sale de la subrutina.
```

```

LATCOMP LDX    LATINS ;Obtiene la diferencia de latitudes para comparar con el
        LDY    LATFIN ;valor límite de acercamiento a coordenada final.
        JSR    RESTA
        CLR    $0004 ;Se hace positivo para facilitar el algoritmo
        LDX    #RESULT ;de comparación.
        JSR    FLTASC
        LDY    #RESULT ;Se verifica el signo de la diferencia de DELTALAT y
        LDY    #LIMITE ;el límite para saber que valor es mayor y poder
        JSR    RESTA ;decidir la acción siguiente.
        LDX    #RESULT
        JSR    FLTASC
        LDAA   $0004 ;Si la comparación es negativa el límite es mayor
        BNE   OFF ;que DELTALAT y el móvil ha llegado a su destino.
        BRA   EXIT ;En caso contrario sale de la subrutina.
OFF      SEI    ;El móvil llegó al destino, deshabilita interrupciones.
        CLR    TMSK1 ;Para apagar el motor de propulsión se modifica la
        CLR    TCTL1 ;condición de toggle por edo. cero en comparación
        RTS    ;exitosa y regresa al programa principal.
EXIT     RTS    ;Si la diferencia entre DELTALONG (o DELTALAT)
        ;y el límite es positiva, termina la subrutina y regresa
        ;al programa principal

```

FIN DE SUBROUTINA LLEGA

CONSTANTE EN COMPARA.ASM

```

        ORG $3FD0
LIMITE: FDB $302E,$3030 ;Límite de Acercamiento
        FDB $3031,$3120 ;que puede modificarse para
        FDB $2020,$2020 ;diferentes distancias en formato ASCCI
        END             ;Valor actual: 0.00011 grados

```

#### 9. TABLA2.ASM

Programa que convierte la tabla de valores tangentes en formato ASCII  
a formato Flotante, se ubica a partir de la dirección \$2B00 hasta la  
dirección \$2B95

DECLARACIÓN DE VARIABLES PARA COLOCAR LOS VALORES FLOTANTES DE LA TABLA

```

TF00    EQU    $2B00 ;Valor de la tangente de cero en flotante (TF00).
TF03    EQU    $2B05 ;Valor de la tangente de tres en flotante (TF03).
TF06    EQU    $2B0A ;Etc.
TF09    EQU    $2B0F
TF12    EQU    $2B14
TF15    EQU    $2B19
TF18    EQU    $2B1E
TF21    EQU    $2B23
TF24    EQU    $2B28
TF27    EQU    $2B2D
TF30    EQU    $2B32
TF33    EQU    $2B37
TF36    EQU    $2B3C
TF39    EQU    $2B41
TF42    EQU    $2B46
TF45    EQU    $2B4B
TF48    EQU    $2B50
TF51    EQU    $2B55

```

```

TF54      EQU      $2B5A
TF57      EQU      $2B5F
TF60      EQU      $2B64
TF63      EQU      $2B69
TF66      EQU      $2B6E
TF69      EQU      $2B73
TF72      EQU      $2B78
TF75      EQU      $2B7D
TF78      EQU      $2B82
TF81      EQU      $2B87
TF84      EQU      $2B8C
TF87      EQU      $2B91

```

9.1 SUBROUTINA DOTAB,  
 CONVIERTE CADA VALOR ASCII DEFINIDO COMO CONSTANTE AL FORMATO FLOTANTE, Y GUARDA  
 ESTOS VALORES FLOTANTES EN UNA NUEVA TABLA DESDE LA DIRECCIÓN \$2B00 A \$2B9B

```

DOTAB      ORG      $01A0      ;Dir. de inicio del programa en memoria.
           LDX      #TAN00     ;Carga el primer valor de tangente ascii en X.
           LDY      #CERO     ;carga la constante CERO en Y y suma X + Y
           JSR      SUMA      ;convirtiendo así el valor ascii a flotante.
           LDX      #TF00     ;Carga dir. destino del valor flotante y
           JSR      PUTFPAC1  ;guarda el valor en esa dirección.
           CLR      $2B04     ;Limpia signo, la tabla es de valores positivos.

           LDX      #TAN03     ;Repite la operación anterior para cada valor de
           LDY      #CERO     ;la tabla de valores ascii (El listado de los
           JSR      SUMA      ;valores restantes se muestra en dos columnas).
           LDX      #TF03
           JSR      PUTFPAC1
           CLR      $2B09

           LDX      #TAN06
           LDY      #CERO
           JSR      SUMA
           LDX      #TF06
           JSR      PUTFPAC1
           CLR      $2B0E
           LDX      #TAN09
           LDY      #CERO
           JSR      SUMA
           LDX      #TF09
           JSR      PUTFPAC1
           CLR      $2B13
           LDX      #TAN12
           LDY      #CERO
           JSR      SUMA
           LDX      #TF12
           JSR      PUTFPAC1
           CLR      $2B18
           LDX      #TAN15
           LDY      #CERO
           JSR      SUMA
           LDX      #TF15
           JSR      PUTFPAC1
           CLR      $2B1D

           LDX      #TAN18
           LDY      #CERO
           JSR      SUMA
           LDX      #TF18
           JSR      PUTFPAC1
           CLR      $2B22
           LDX      #TAN21
           LDY      #CERO
           JSR      SUMA
           LDX      #TF21
           JSR      PUTFPAC1
           CLR      $2B27
           LDX      #TAN24
           LDY      #CERO
           JSR      SUMA
           LDX      #TF24
           JSR      PUTFPAC1
           CLR      $2B2C
           LDX      #TAN27
           LDY      #CERO
           JSR      SUMA
           LDX      #TF27
           JSR      PUTFPAC1
           CLR      $2B31

```



LDX #TAN30  
LDY #CERO  
JSR SUMA  
LDX #TF30  
JSR PUTFPAC1  
CLR \$2B36  
LDX #TAN33  
LDY #CERO  
JSR SUMA  
LDX #TF33  
JSR PUTFPAC1  
CLR \$2B3B  
LDX #TAN36  
LDY #CERO  
JSR SUMA  
LDX #TF36  
JSR PUTFPAC1  
CLR \$2B40  
LDX #TAN39  
LDY #CERO  
JSR SUMA  
LDX #TF39  
JSR PUTFPAC1  
CLR \$2B45  
LDX #TAN42  
LDY #CERO  
JSR SUMA  
LDX #TF42  
JSR PUTFPAC1  
CLR \$2B4A  
LDX #TAN45  
LDY #CERO  
JSR SUMA  
LDX #TF45  
JSR PUTFPAC1  
CLR \$2B4F  
LDX #TAN48  
LDY #CERO  
JSR SUMA  
LDX #TF48  
JSR PUTFPAC1  
CLR 2B54  
LDX #TAN51  
LDY #CERO  
JSR SUMA  
LDX #TF51  
JSR PUTFPAC1  
CLR \$2B59  
LDX #TAN54  
LDY #CERO  
JSR SUMA  
LDX #TF54  
JSR PUTFPAC1  
CLR \$2B5E

LDX #TAN57  
LDY #CERO  
JSR SUMA  
LDX #TF57  
JSR PUTFPAC1  
CLR \$2B63  
LDX #TAN60  
LDY #CERO  
JSR SUMA  
LDX #TF60  
JSR PUTFPAC1  
CLR \$2B68  
LDX #TAN63  
LDY #CERO  
JSR SUMA  
LDX #TF63  
JSR PUTFPAC1  
CLR \$2B6D  
LDX #TAN66  
LDY #CERO  
JSR SUMA  
LDX #TF66  
JSR PUTFPAC1  
CLR \$2B72  
LDX #TAN69  
LDY #CERO  
JSR SUMA  
LDX #TF69  
JSR PUTFPAC1  
CLR \$2B77  
LDX #TAN72  
LDY #CERO  
JSR SUMA  
LDX #TF72  
JSR PUTFPAC1  
CLR \$2B7C  
LDX #TAN75  
LDY #CERO  
JSR SUMA  
LDX #TF75  
JSR PUTFPAC1  
CLR \$2B81  
LDX #TAN78  
LDY #CERO  
JSR SUMA  
LDX #TF78  
JSR PUTFPAC1  
CLR \$2B86  
LDX #TAN81  
LDY #CERO  
JSR SUMA  
LDX #TF81  
JSR PUTFPAC1  
CLR \$2B8B

```

LDX #TAN84
LDY #CERO
JSR SUMA
LDX #TF84
JSR PUTFPAC1
CLR $2B90

```

```

LDX #TAN87
LDY #CERO
JSR SUMA
LDX #TF87
JSR PUTFPAC1
CLR $2B95
RTS

```

;Termina.

FIN DE SUBROUTINA DOTAB

CONSTANTES EN DOTAB.ASM

```

ORG $03E8 ;Se ocupa para pasar de ascii a flotante sumando cero
CERO: FDB $3030,$3030 ;en ascii a otro número ascii, el resultado de la suma
      FDB $3030,$2020 ;queda en flotante en FPAC1.

```

TABLA DE VALORES EN FORMATO ASCII (Debe estar en una sola columna)

|        |                   |                          |
|--------|-------------------|--------------------------|
| TAN00: | ORG \$03F0        | FDB \$3730,\$2020        |
|        | FDB \$302E,\$3030 | ORG \$0458               |
|        | FDB \$3030,\$2020 | TAN39: FDB \$302E,\$3831 |
|        | ORG \$03F8        | FDB \$3030,\$2020        |
| TAN03: | FDB \$302E,\$3035 | ORG \$0460               |
|        | FDB \$3230,\$2020 | TAN42: FDB \$302E,\$3930 |
|        | ORG \$0400        | FDB \$3030,\$2020        |
| TAN06: | FDB \$302E,\$3130 | ORG \$0468               |
|        | FDB \$3530,\$2020 | TAN45: FDB \$312E,\$3030 |
|        | ORG \$0408        | FDB \$3030,\$2020        |
| TAN09: | FDB \$302E,\$3135 | ORG \$0470               |
|        | FDB \$3830,\$2020 | TAN48: FDB \$312E,\$3131 |
|        | ORG \$0410        | FDB \$3130,\$2020        |
| TAN12: | FDB \$302E,\$3231 | ORG \$0478               |
|        | FDB \$3330,\$2020 | TAN51: FDB \$312E,\$3233 |
|        | ORG \$0418        | FDB \$3530,\$2020        |
| TAN15: | FDB \$302E,\$3236 | ORG \$0480               |
|        | FDB \$3830,\$2020 | TAN54: FDB \$312E,\$3337 |
|        | ORG \$0420        | FDB \$3630,\$2020        |
| TAN18: | FDB \$302E,\$3332 | ORG \$0488               |
|        | FDB \$3530,\$2020 | TAN57: FDB \$312E,\$3534 |
|        | ORG \$0428        | FDB \$3030,\$2020        |
| TAN21: | FDB \$302E,\$3338 | ORG \$0490               |
|        | FDB \$3430,\$2020 | TAN60: FDB \$312E,\$3733 |
|        | ORG \$0430        | FDB \$3230,\$2020        |
| TAN24: | FDB \$302E,\$3434 | ORG \$0498               |
|        | FDB \$3530,\$2020 | TAN63: FDB \$312E,\$3730 |
|        | ORG \$0438        | FDB \$3530,\$2020        |
| TAN27: | FDB \$302E,\$3531 | ORG \$04A0               |
|        | FDB \$3030,\$2020 | TAN66: FDB \$322E,\$3234 |
|        | ORG \$0440        | FDB \$3630,\$2020        |
| TAN30: | FDB \$302E,\$3537 | ORG \$04A8               |
|        | FDB \$3730,\$2020 | TAN69: FDB \$322E,\$3630 |
|        | ORG \$0448        | FDB \$3530,\$2020        |
| TAN33: | FDB \$302E,\$3634 | ORG \$04B0               |
|        | FDB \$3930,\$2020 | TAN72: FDB \$332E,\$3037 |
|        | ORG \$0450        | FDB \$3830,\$2020        |
| TAN36: | FDB \$302E,\$3732 | ORG \$04B8               |

TAN75: FDB \$332E,\$3733  
 FDB \$3230,\$2020  
 ORG \$04C0  
 TAN78: FDB \$342E,\$3730  
 FDB \$3530,\$2020  
 ORG \$04C8  
 TAN81: FDB \$362E,\$3331  
 FDB \$3430,\$2020

ORG \$04D0  
 TAN84: FDB \$392E,\$3531  
 FDB \$3430,\$2020  
 ORG \$04D8  
 TAN87: FDB \$3139,\$2E30  
 FDB \$3830,\$2020  
 END

### 10. GD1.ASM

Este programa guarda de una localidad de memoria a otra

#### 10.1 SUBROUTINA GDCOORD1, GUARDA UNA CADENA DE DATOS DE 15 CARACTERES DE LONGITUD

```

      ORG    $3FFF      ;Dir. de inicio del programa en RAM.
GDCOORD1 LDAB  #SOF    ;Carga 0Fh = 15d en B, como contador.
GUARDA1 LDAA  0,X      ;Carga en A el contenido de X (se espera una dirección).
          STAA  0,Y      ;Guarda el valor de A en la dir. indicada por Y.
          INX          ;Se incrementa X al siguiente valor (sig. Carácter).
          INY          ;Se incrementa Y a la siguiente dirección de destino.
          DECB        ;Se decrementa B, contador de caracteres guardados.
          BNE  GUARDA1  ;Mientras B!=0 repite la operación de guardar.
          RTS          ;Cuando B = 0, termina la subrutina.

```

FIN DE SUBROUTINA GDCOORD1