

37
Ej



UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO
FACULTAD DE INGENIERÍA

**SISTEMA AEROPORTADO PARA LA
ADQUISICIÓN DE IMÁGENES DE
VIDEO**

TESIS



QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN
PRESENTA
RUBÉN DOMÍNGUEZ TREJO

Directora: M. en C. Margarita Navarrete Montesinos

Laboratorio de Ingeniería Aeroespacial
Instituto de Ingeniería
U.N.A.M. México, D.F. 1996

**TESIS CON
FALLA DE ORIGEN**

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



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.

ÍNDICE.

AGRADECIMIENTOS.	3
RESUMEN.	4
1. INTRODUCCIÓN.	5
1.1 Antecedentes.	5
1.2 Percepción remota.	6
1.3 Videogrametría.	9
2. DESCRIPCIÓN DEL EQUIPO UTILIZADO.	11
2.1 Microcomputadora.	13
2.2 Cámaras de video.	14
2.3 Receptor del sistema de posicionamiento global (GPS).	15
2.4 Contenedor de cámaras de video.	18
2.4.1 Mecanismo de posicionamiento y ajuste de distancia focal.	21
2.5 Sensores y servomotores.	22
3. ELECTRÓNICA DE CONTROL Y DE ADQUISICIÓN DE DATOS.	26
3.1 Generalidades.	26
3.2 Decodificación de direcciones.	27
3.3 Control de funciones de las cámaras de video.	28
3.4 Control del mecanismo de posicionamiento y ajuste de distancia focal.	30
3.5 Adquisición de datos de sensores.	34
3.6 Etiquetación de imágenes.	36
4. PROGRAMA DE CONTROL.	40
4.1 Generalidades.	40
4.2 Modo de operación en Vuelo.	42
4.2.1 Manejo del receptor GPS.	45
4.2.2 Manejo de Cámaras.	48
4.3 Modo de operación en Tierra.	51

5. PRUEBAS EXPERIMENTALES Y DISCUSIÓN DE RESULTADOS.	56
5.1 Pruebas al control de las funciones de las cámaras.	56
5.2 Pruebas de etiquetación de imágenes.	56
5.3 Pruebas al mecanismo de posicionamiento y ajuste de distancia focal.	58
5.4 Pruebas a la adquisición de datos de sensores.	58
5.5 Pruebas de manejo del receptor GPS.	59
5.6 Discusión de resultados.	60
6. CONCLUSIONES Y RECOMENDACIONES.	61
APÉNDICE A: CARACTERÍSTICAS DEL RECEPTOR GPS.	64
APÉNDICE B: CARACTERÍSTICAS DE LA PISTA DE AUDIO DE LAS CINTAS.	66
APÉNDICE C: EL PLL (PHASE LOCKED LOOP).	68
APÉNDICE D: LENGUAJES DE PROGRAMACIÓN.	71
APÉNDICE E: CÓDIGO DEL PROGRAMA DE CONTROL.	73
REFERENCIAS.	92

AGRADECIMIENTOS

A la M. en C. Margarita Navarrete Montesinos, por su ayuda y orientación.

Al M. en I. Jorge Prado Molina, por sus valiosos comentarios y asesorías.

Al Dr. Ricardo Peralta y Fabi, jefe del Laboratorio de Ingeniería Aeroespacial del Instituto de Ingeniería, por las facilidades y el apoyo brindados para la elaboración de esta tesis.

RESUMEN

Se presenta el diseño y configuración de una serie de interfaces dedicadas al control de un sistema aeroportado para la adquisición de imágenes de video, dichas imágenes se utilizan en diferentes aplicaciones de percepción remota.

Se diseñó y construyó la electrónica para el control de servomotores, la adquisición de datos de sensores de orientación espacial y apuntamiento, y la activación de funciones propias de cámaras de video tipo CCD. La electrónica es activada y supervisada por medio de un programa en lenguaje C, mediante una microcomputadora. Finalmente, se muestran los diagramas electrónicos y de programación así como los resultados de pruebas de laboratorio para las diferentes interfaces, asimismo, se incluyen recomendaciones para ampliar y mejorar este sistema [6].

1. INTRODUCCIÓN

1.1 Antecedentes.

Dada la importancia actual de las imágenes obtenidas desde satélites y aeronaves para realizar diversos estudios de percepción remota, tales como: rendimiento agrícola, actividad pesquera, diseño de carreteras, detección de incendios, etc., y dado el elevado costo y tiempo de adquisición, así como la dependencia para obtener este servicio de compañías extranjeras (sobre todo en imágenes de satélite), el Laboratorio de Ingeniería Aeroespacial del Instituto de Ingeniería de la UNAM se ha propuesto desarrollar sistemas de percepción remota propio; cuyo objetivo principal es obtener imágenes de video y fotográficas del territorio nacional, a un costo bajo y en tiempos cortos [1].

Un estudio realizado sobre el uso de equipos y conceptos alternativos para realizar percepción remota [12], basados en técnicas al alcance de Institutos, Centros de investigación y de Desarrollo en México, indicó la factibilidad y ventajas de realizar el proyecto en el país. El proyecto denominado Aeronave a Control Remoto (ACR), consiste en el diseño, construcción, pruebas y operación de una aeronave de control remoto, cuyo objetivo principal es la teledetección es decir; la adquisición de diferentes tipos de imágenes (fotográficas, de video, infrarrojas, etc.). Asimismo, contempla el estudio de diversas aplicaciones como: muestreo y análisis del aire, magnetometría, radar de apertura sintética, exploración de minerales estratégicos, entre otras [2]. Una de las etapas que compone este proyecto se desarrolla en esta tesis.

El Laboratorio de Ingeniería Aeroespacial ha venido trabajando desde hace tiempo sobre percepción remota para realizar estudios diversos del territorio nacional, a través de imágenes multiespectrales (imágenes tomadas en bandas bien delimitadas del espectro electromagnético). Esta información permite analizar diferentes extensiones de terreno para conocer sus atributos y

características, haciendo posible la toma de decisiones racionales sobre diferentes aspectos de la organización territorial, tales como infraestructura, ordenamiento, uso de suelo, tipo y extensión de cultivos, pérdida de suelo por procesos de erosión, explosión demográfica, afloramiento de minerales, contaminación en cuerpos de agua y en suelos, entre muchos más.

1.2 Percepción remota.

La percepción remota se interesa por la medición de la reflexión de la radiación electromagnética de la superficie del planeta. Alguna o varias porciones del espectro electromagnético pueden estar involucradas. Las bandas visible, infrarroja y de radar, son las más utilizadas.

La utilidad de una imagen y la habilidad de derivar características específicas sobre ellas depende de varios factores como son :

- **Contraste;** intervalo de brillo de la escena.
- **Tono;** distinguir matices del negro al blanco.
- **Textura;** frecuencia con la cual cambian los tonos.
- **Detectabilidad;** registra la presencia o ausencia de una determinada característica.
- **Reconocibilidad;** habilidad de identificar un objeto por su apariencia.
- **Firma;** es la definición de ciertas combinaciones de características espectrales que permiten a un objeto ser clasificado, las firmas se establecen empíricamente examinando el terreno y realizando asociaciones con lo contenido en las imágenes. La diferencia entre firmas proporciona las bases para el proceso de clasificación y análisis de las imágenes.

La fotografía ha sido la primera forma de obtención de imágenes para Percepción Remota. La resolución de las películas actuales está limitada por factores ópticos y de tiempo de exposición, más que por el medio de grabación. Las imágenes obtenidas desde gran altura permiten percibir lineamientos y características no reconocidas en otras circunstancias, debido a la perspectiva sinóptica que se tiene. Las imágenes con una gran cobertura permiten hacer estudios a gran escala.

Numerosas características son reveladas cuando las áreas de estudio son fotografiadas en diferentes formas, ubicaciones, bandas espectrales, direcciones, ángulos solares, estaciones del año, horas del día, elevaciones y polarizaciones. De la combinación de varias imágenes se obtiene mas información que de una sola. La fotografía aérea debe realizarse de cinco a seis horas alrededor del medio día, cuando el ángulo solar es alto y el área de sombras es mínimo. Sin embargo, algunas veces conviene fotografiar con un ángulo solar bajo para resaltar relieves.

Las fotografías aéreas generalmente se obtienen con un traslape del 60% entre imágenes sucesivas, así observando pares de fotografías de la misma zona desde diferentes puntos de toma, se produce un efecto tridimensional virtual donde las elevaciones pueden ser determinadas por paralelismo. Las imágenes infrarrojas permiten enfatizar diferencias asociadas con vegetación y agua. El color es especialmente valioso, ya que el ojo distingue un mayor número de colores que de tonos de gris.

Una vez realizada la toma de imágenes se procede a su análisis. El procesamiento de imágenes se realiza fundamentalmente para corregir errores de grabación, distorsiones geométricas, combinar imágenes, resaltar características, sintetizar pares de imágenes para facilitar el estudio del relieve y la clasificación de patrones.

Durante la obtención de las imágenes, uno de los principales problemas que se enfrenta es el de la distorsión geométrica que es causada por la óptica de la cámara, por la distancia variable entre la superficie y la aeronave y la variación del centro de toma, causada por inestabilidades del vuelo (alabeo, cabeceo principalmente). Estos efectos no son lineales, pero se pueden corregir si se conocen los movimientos ocurridos durante la adquisición. Con las coordenadas del centro de toma y el vector de apuntamiento es posible crear mosaicos de la zona específica y además corregir y ajustar la escala al unir las imágenes.

Para obtener datos de la perturbación del contenedor de las cámaras y llevar a cabo las correcciones pertinentes, éste ha sido instrumentado con diferentes sensores para obtener información de la inclinación respecto al plano horizontal, la aceleración, la dirección de vuelo, la posición (X, Y, Z) y la hora. Todos estos parámetros son adquiridos durante el vuelo y almacenados para realizar el procesamiento posterior en tierra. El almacenamiento de éstos datos, así como el control y monitoreo del sistema de adquisición de imágenes constituyen el objetivo central de este trabajo de tesis.

Las imágenes así obtenidas son utilizadas en la actualización de mapas existentes que no presentan el detalle suficiente o son poco precisos, como son los de áreas poco accesibles. La exploración minera utiliza la percepción remota para mapear características tales como sistemas de fracturas y alteraciones de la superficie. También existen otros muchos usos de la percepción remota, tales como la detección de agua subterránea, de tuberías, contaminación, movimientos de témpanos, inundaciones, efectos de tormentas, erosión etc. Las imágenes tienen inclusive aplicaciones como: estudios meteorológicos, localización de plagas, ubicación de lugares arqueológicos, etc.

1.3 Videogrametría.

La aparición de las cámaras de video con sensor de estado sólido CCD introduce al equipo televisivo en la tarea de teledetección, que no se había podido lograr antes por las distorsiones inherentes de las cámaras con vidicon.

La combinación de este tipo de cámaras CCD en conjunto con filtros de interferencia en sectores bien delimitados del espectro electromagnético, dió lugar a una nueva tecnología que se ha denominado en concordancia con la literatura internacional: videogrametría multiespectral [3].

Los filtros más utilizados frente al sistema óptico se seleccionan con base en las características del terreno y sus propiedades, a saber:

FILTRO	PICO (nm)	ANCHO DE BANDA (nm)	TRASMITANCIA (%)
Azul	452	110	70
Verde	550	100	66
Rojo	655	150	56

Otros filtros manejados son los del infrarrojo cercano, con centros en los 750 y en los 850 nanómetros. Los filtros rojo, verde y azul han demostrado ser suficientes para la mayoría de las misiones que se han llevado a cabo en la banda visible, sin embargo, para cada misión es necesario realizar pruebas preliminares para determinar cuales son los filtros más adecuados, dependiendo del problema específico que se quiera abordar.

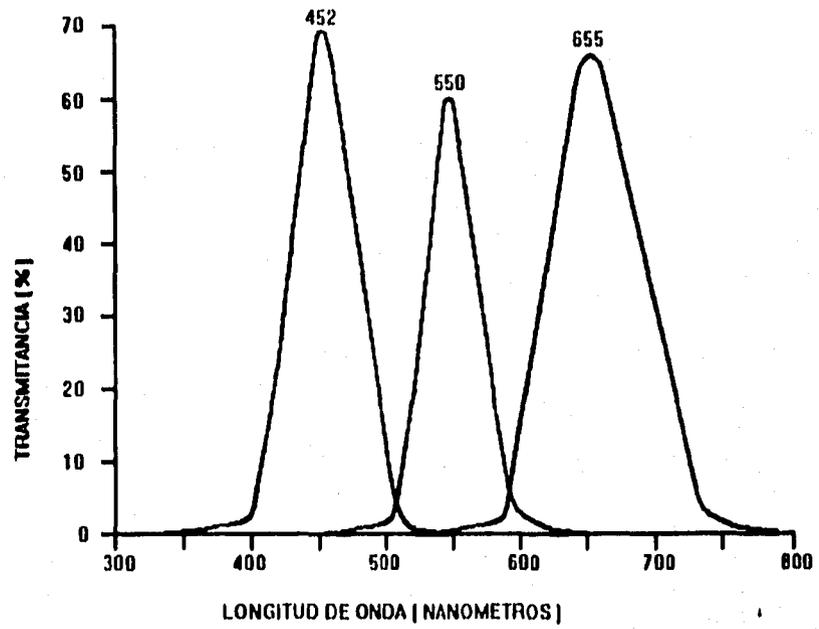


Figura 1. Gráfica de Transmitancia de los filtros azul, verde y rojo.

2. DESCRIPCIÓN DEL EQUIPO UTILIZADO.

La elección del equipo utilizado siguió criterios centrales tales como: bajo costo, alta fiabilidad de funcionamiento y fácil acceso en el mercado nacional, para asegurar la duplicación del sistema por cualquier grupo interesado, con la asesoría del personal del Laboratorio de Ingeniería Aeroespacial del Instituto de Ingeniería.

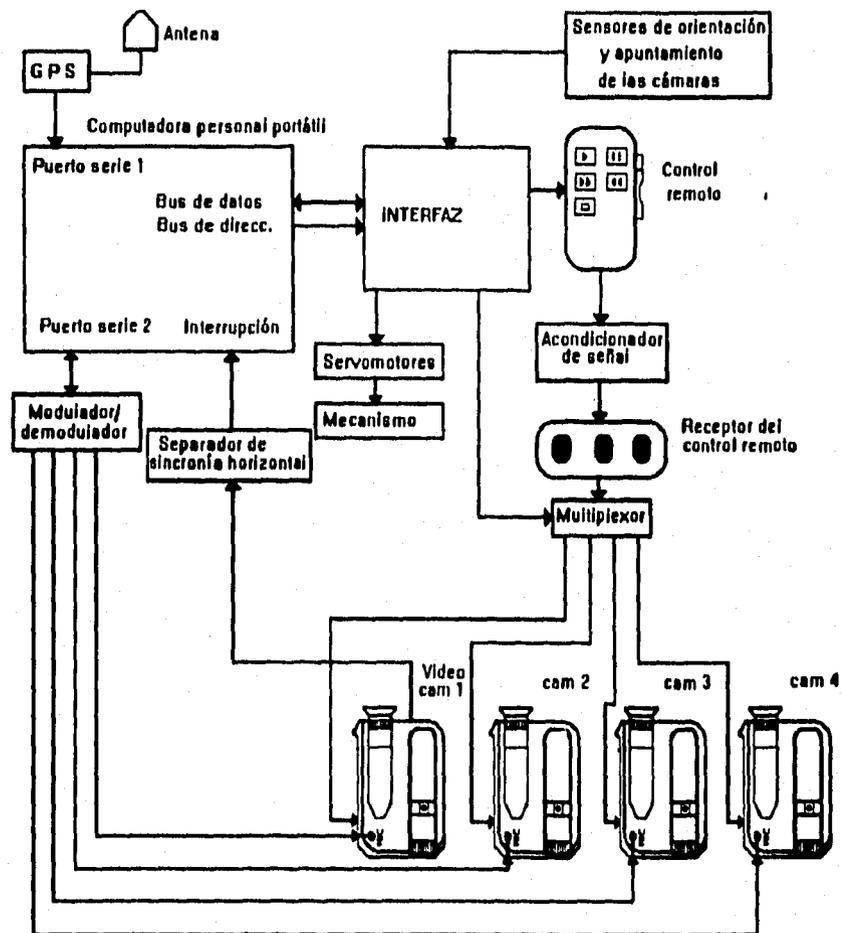


Figura 2. Diagrama a bloques del sistema de adquisición de imágenes.

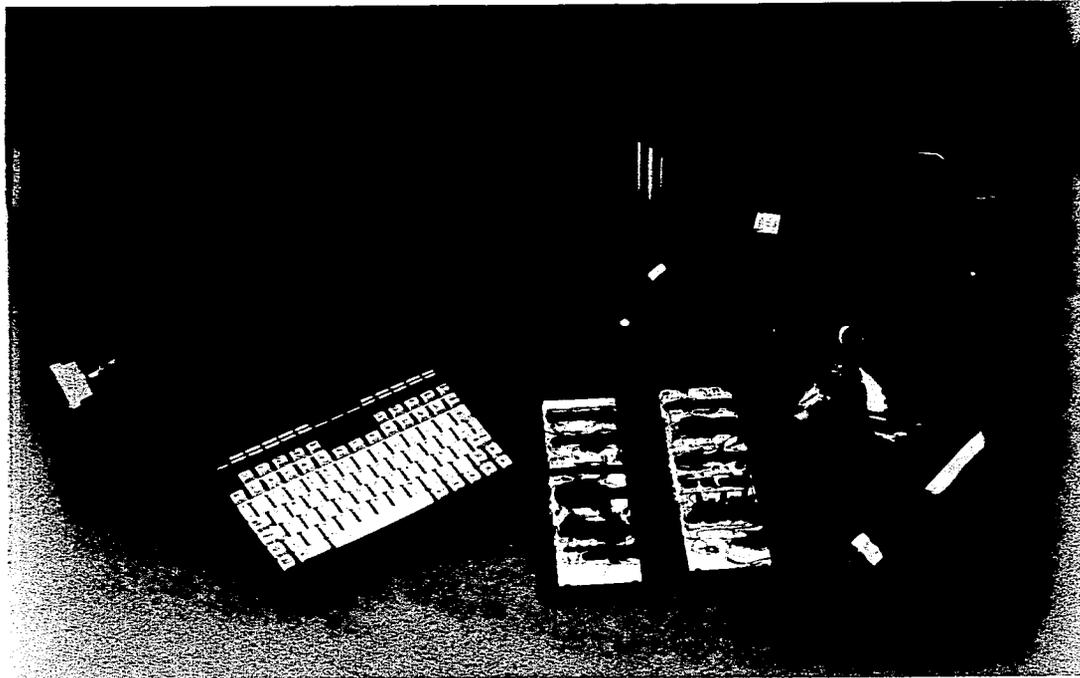


Figura 3. Fotografía de la computadora, la caja de expansión, receptor GPS, módem FSK y prototipo de las interfaces del sistema.

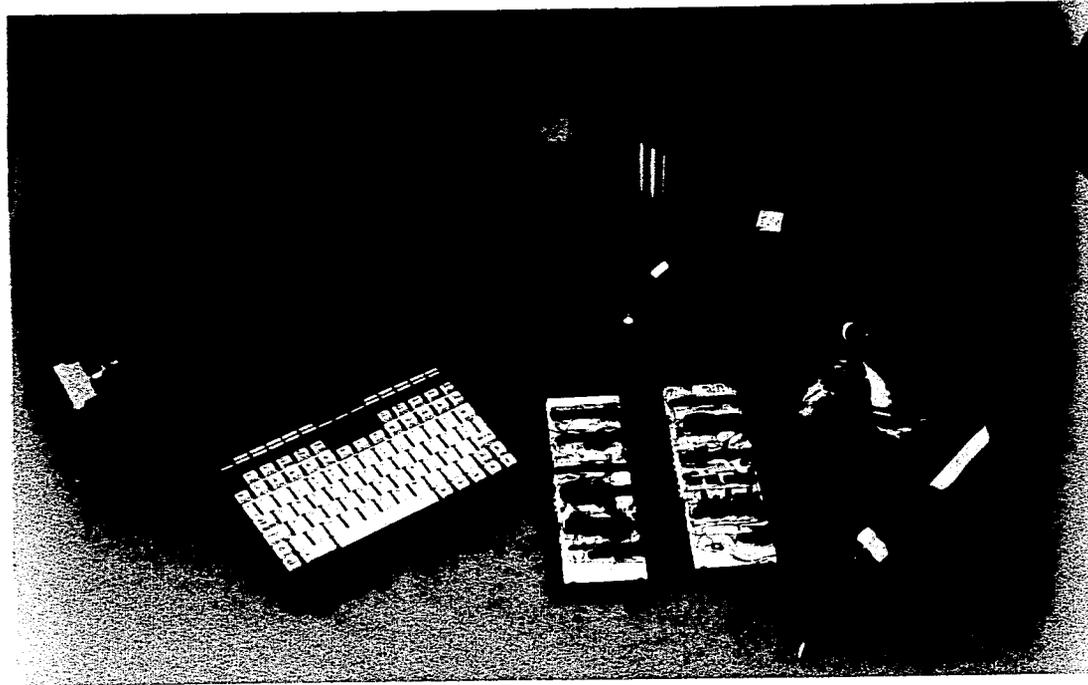


Figura 3. Fotografía de la computadora, la caja de expansión, receptor GPS, módem FSK y prototipo de las interfaces del sistema.

Por medio de una computadora y a través de diferentes interfaces se controlan todos los componentes del sistema, a saber:

- Se controlan hasta cuatro cámaras de video a través de un control remoto.
- Se adquieren datos de hasta 16 sensores.
- Se manipula la orientación y distancia focal de una de las cámaras.
- Se monitorean constantemente las cámaras y sensores disponibles en el sistema, para asegurar su funcionamiento.
- Al través de puerto serie se establece comunicación con el receptor de posición llamada GPS, a la vez que se verifica también su funcionamiento.
- Por el puerto serie y mediante un circuito acondicionador se graba una etiqueta de identificación en cada cuadro tomado por las cámaras de video. Para tal efecto, se requiere sincronizar dichas tomas, lo que se realiza aprovechando los servicios de interrupción de la computadora.

2.1. Microcomputadora.

Se utiliza una computadora AT compatible, basada en el microprocesador Intel 386, con una masa de 3.3 kilogramos, velocidad de reloj de 20 MHz, conmutable a 4 MHz en períodos de inactividad, fundamental para el ahorro de energía. Pantalla de cristal líquido de iluminación de fondo, que permite una adecuada nitidez en un ambiente de operación rudo y con un bajo consumo de potencia. Baterías de níquel-cadmio permiten el funcionamiento continuo de la computadora hasta por dos horas.

Cuenta también con dos puertos serie RS232, un puerto paralelo, un puerto para monitor externo, un conector de 100 terminales para una caja de expansión.

La caja de expansión incluye dos ranuras AT y una ranura XT, y alimentación de energía independiente. Debido a la utilización del equipo a bordo en aeronaves, la caja de expansión se energiza por medio de un banco de baterías.

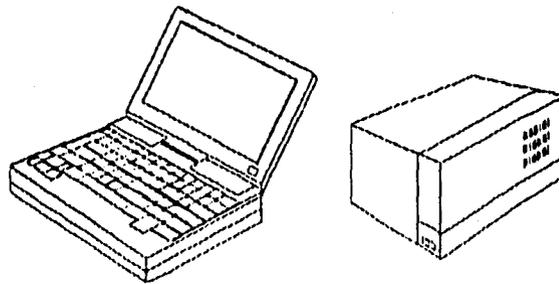


Figura 4. Computadora y caja de expansión utilizadas en el sistema.

2.2 Cámaras de Video.

Las cámaras de video fueron seleccionadas siguiendo varios criterios: costo bajo de adquisición, estabilidad de imagen, confiabilidad alta, tolerancia al uso relativamente rudo, versatilidad, peso bajo y consumo de energía bajo.

Las cámaras elegidas presentan las siguientes características: manejo fácil, grabadora integrada, distancia focal ajustable, tiempo de exposición ajustable hasta 1/4000 de segundo, velocidad de grabación estándar de 30 imágenes por segundo, peso bajo y consumo eléctrico bajo. Las cintas de video que utiliza son de 8 mm. Por ser de uso general son capaces de soportar trato relativamente rudo. Siendo cámaras CCD, poseen estabilidad geométrica alta, ya que, si varias cámaras tienen sus ejes ópticos paralelos a un eje común, las cámaras captarán la misma zona en el cuadro adquirido simultáneamente.

A éstas cámaras se les adaptan filtros que permiten tomar imágenes en diferentes bandas del espectro electromagnético, principalmente azul, verde, rojo e infrarrojo cercano hasta 850 nanómetros, como se ha mencionado.

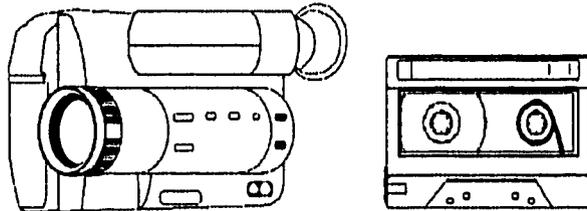


Figura 5. Cámara de video y video cinta utilizadas en el sistema

2.3 Receptor del Sistema de Posicionamiento Global.

El sistema de posicionamiento global (GPS, Global Positioning System) es un sistema de posicionamiento y navegación por tierra, mar y aire basado en la interpretación de la emisión de señales de un grupo de satélites y recibidas por receptores portátiles en tierra, o como en este caso en aeronaves.

El GPS no es el primer sistema de radio navegación, ni tampoco es el primero que tenga cobertura mundial como sistema de navegación basado en satélites; sin embargo, fue creado con base en uno de los sistemas existentes adicionándole desde luego mejoras importantes.

El sistema GPS consta de tres segmentos: el segmento espacial, el de control y el del usuario. El segmento espacial consta de un grupo de 24 satélites que componen el sistema GPS, 21 de los cuales están en operación y 3 son de reserva. Los satélites operan en órbitas circulares a 20,200 kilómetros sobre la Tierra, y están ordenados en seis planos orbitales, separados 55 grados entre sí. Los satélites tienen un período orbital de 12 horas, aproximadamente. Su

disposición asegura que en cualquier momento al menos 5 satélites estarán en línea de vista del usuario en cualquier lugar del mundo. Un mínimo de cuatro satélites se requieren para el posicionamiento de precisión.

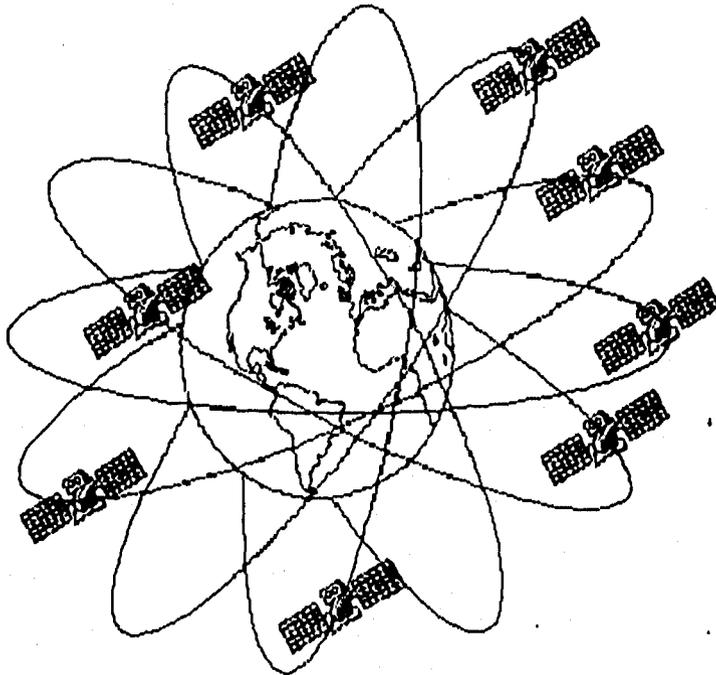


Figura 6. Sistema de Posicionamiento Global. Segmento espacial.

El segmento de control consta de una estación maestra de control en Colorado Springs, CO., EUA, cinco estaciones de monitoreo y tres antenas terrestres ubicadas alrededor del planeta. Las estaciones de monitoreo rastrean todos los satélites a la vista, y envían los datos acumulados a la estación de control maestro, el cual calcula con gran precisión las órbitas de los satélites. Cualquier actualización de la posición es transmitida a cada uno de los satélites a través de las antenas terrestres.

El segmento del usuario consta de antenas y receptores que pueden operar transportados por tierra, mar o aire, para determinar su posición y velocidad. El dato de la hora, también se puede obtener.

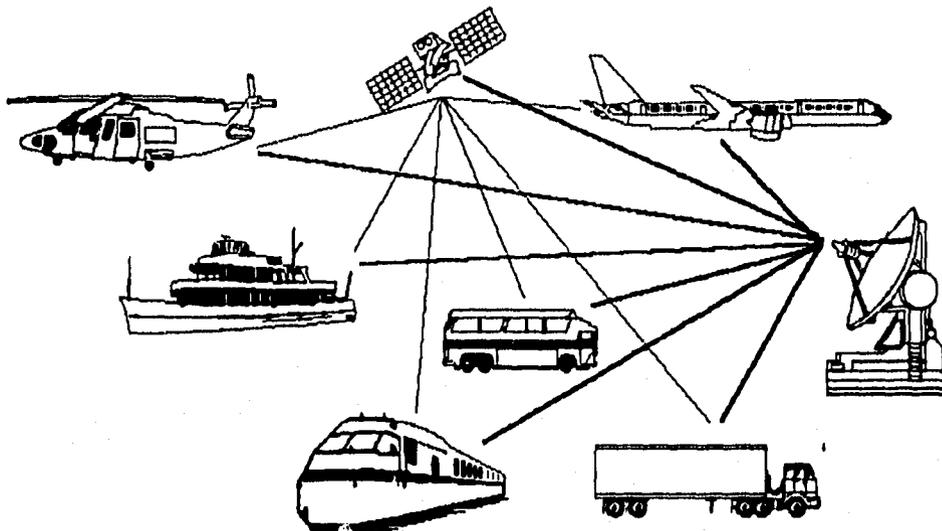


Figura 7. Sistema de Posicionamiento Global. Segmento del usuario.

Aún cuando los errores causados por irregularidades de propagación de las señales de los satélites son considerables, la mayor fuente de error es deliberadamente añadida a la señal del GPS por el Departamento de Defensa de los Estados Unidos. La razón por la que se diseñó éste modo de operación, llamado de disponibilidad selectiva (S/A, Selective Available), fue para hacer menos preciso el sistema a usuarios no autorizados.

Los satélites GPS transmiten dos señales. La primera es de adquisición ordinaria (C/A, Coarse Acquisition) recibida por los receptores civiles. La segunda señal es de precisión (P, Precision).

La señal P está encriptada y sólo los usuarios autorizados tienen acceso a ella. La señal C/A no está encriptada, pero el modo S/A degrada la precisión inherente de 20 ó 30 metros hasta 100 metros o más. Existe sin embargo el GPS funcionando en modo diferencial, lo que permite determinar con precisión la posición con este método, si se conoce la localización exacta de un receptor fijo se pueden determinar qué errores están siendo introducidos al satélite, y procesar los datos para filtrar los errores.

En fechas recientes INMARSAT, un operador internacional de satélites, ha señalado su intención de competir con los servicios norteamericano y ruso de posicionamiento global por satélite (GPS y GLONASS, respectivamente), introduciendo mejoras a los servicios potenciales del GPS, con su siguiente generación de satélites Inmarsat-3.

INMARSAT sostiene que los nuevos satélites serían lanzados a finales de 1995 o principios de 1996, éstos ofrecerán una mayor precisión en los servicios de posicionamiento que la proporcionada por los servicios existentes.

La organización de satélites dice que la corrección de señales de Inmarsat-3 enclavará la posición de los usuarios dentro de los 10 m, comparado con los 50 m comúnmente obtenidos por el uso del servicio norteamericano.

Con la información proporcionada por el receptor GPS, se puede establecer el centro de toma de las imágenes grabadas durante el vuelo de la aeronave, y así conocer la trayectoria real seguida durante la adquisición.

2.4 Contenedor de cámaras de video.

Se emplea un contenedor aeroportado desarrollado en el Laboratorio de Ingeniería Aeroespacial, para realizar tareas diversas de percepción remota en forma económica y adaptable a las necesidades de cada caso. El equipo adquiere

imágenes de video en la región visible del espectro electromagnético, con la posibilidad de ajustar el área rastreada y la resolución variando la altura de vuelo o la distancia focal de las cámaras.

La carga útil del contenedor consiste en: cuatro cámaras de video, una de ellas con distancia focal variable y con control de apuntamiento y las otras tres son fijas pero pueden traslapar su cobertura hasta un 100%; un sensor de orientación que proporciona la medida del ángulo del contenedor respecto al Norte; dos sensores de inclinación para conocer el vector de apuntamiento de las cámaras, y espacio suficiente para ubicar otros sensores.

El contenedor proporciona las siguientes ventajas: asegura la carga útil, tiene los aditamentos necesarios para colocar, ajustar y alinear rápidamente las cámaras de video, es de fácil acceso para maniobras de calibración y ajuste, aísla la carga útil de las vibraciones de la aeronave, y es de fácil adaptación al tipo de sujeción que presente el fuselaje de la aeronave sin distorsionar apreciablemente su perfil aerodinámico [4].

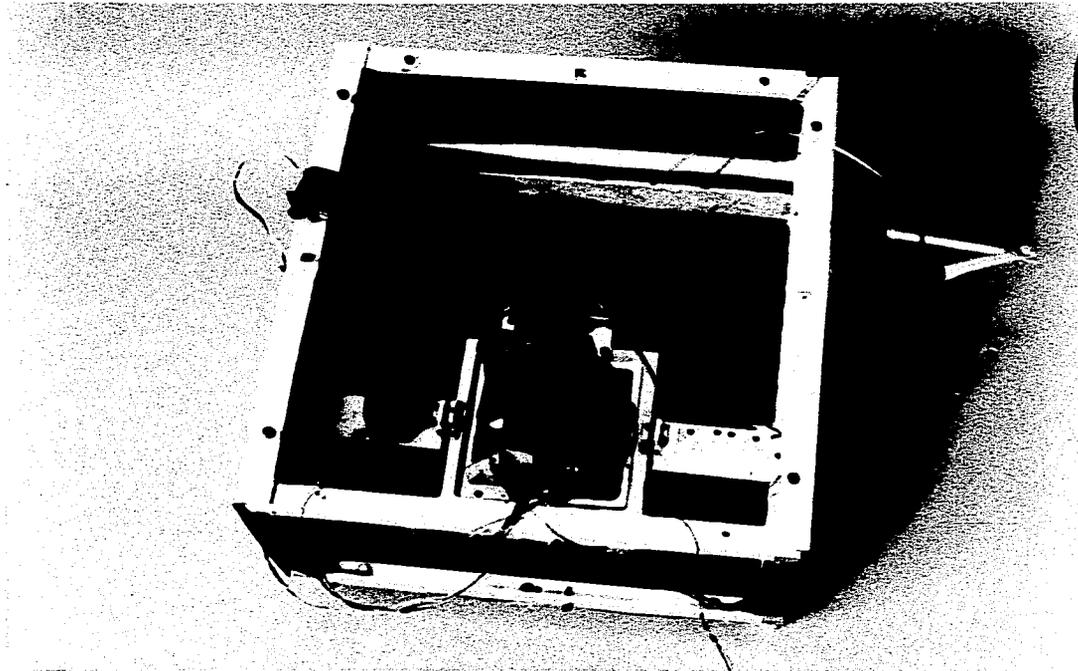


Figura 8. Fotografía del contenedor.

2.4.1 Mecanismo de posicionamiento y ajuste de distancia focal.

El aditamento para el control de apuntamiento de la cámara móvil consta de dos marcos concéntricos pivoteados, la rotación de cada marco es llevada a cabo por servomotores independientes, lo que permite un apuntamiento rápido. Además se acondicionó un mecanismo con cable y servomotor para manipular la distancia focal de la cámara móvil. La utilidad de esta cámara se pone de manifiesto al momento de interpretar las imágenes adquiridas, ya que el acercamiento sobre sitios o rasgos específicos sobre el terreno permite la identificación de éstos sin necesidad de llevar a cabo visitas de campo, con el consecuente ahorro de tiempo en traslados.

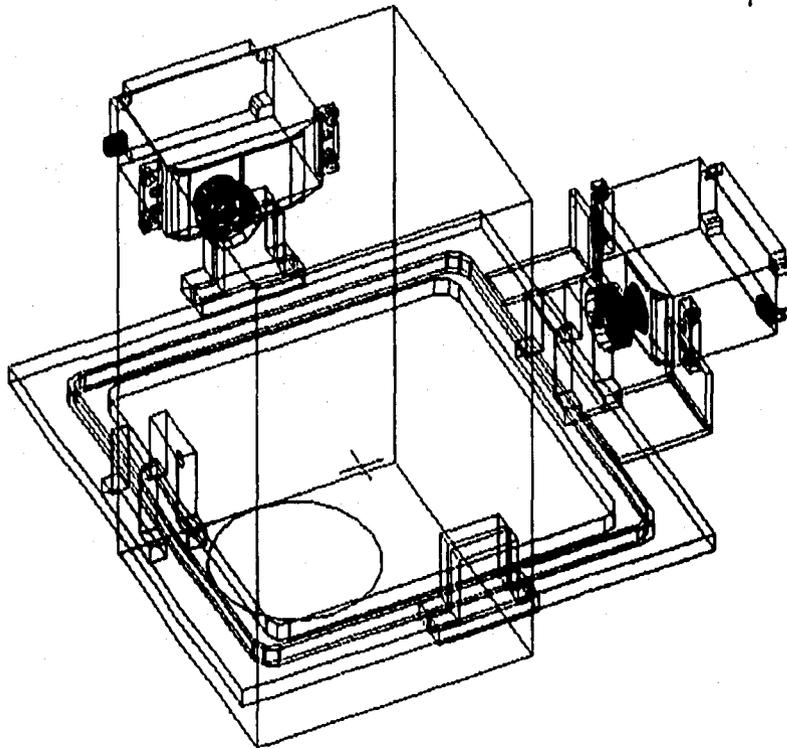


Figura 9. Dibujo del mecanismo de control de apuntamiento de la cámara móvil.

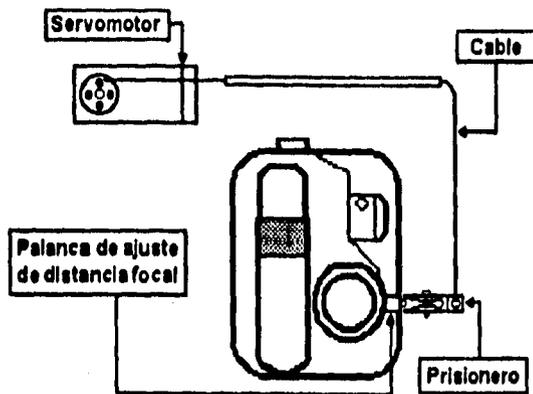


Figura 10. Dibujo del mecanismo para variar la distancia focal.

2.5 Sensores y Servomotores

A) Altitud.

Operación: Su magnitud se estima a partir de la medición de la temperatura y la presión del ambiente. Con estos datos se obtiene una densidad del aire, que se compara con los valores de la atmósfera estándar.

	Presión	Temperatura
Intervalo	0 a 1.5 bar	-25 a 85 °C
Error:	0.25%	± 10 °C
Sensibilidad:	66.011 mV / bar	10 mV / °C
Máxima no linealidad	± 0.25%	2%

Sensor de Presión.

Operación: Es un diafragma que al moverse por la presión del ambiente ejerce un esfuerzo sobre una resistencia, la cual cambia su magnitud de manera proporcional al cambio de presión.

Sensor de Temperatura.

Operación: Es un transistor, el voltaje base-emisor varía linealmente respecto al cambio de temperatura.

Los valores de densidad contra altitud están referidos a una atmósfera ideal (Atmósfera Normal Internacional), cuyas características son las siguientes:

Temperatura:	$T_0 = 273 + 15.3 = 288.3\text{K}$
Presión (snm):	$P_0 = 760 \text{ mm de Hg}$
Densidad:	$\rho_0 = 0.12497 \text{ kg / m}^3$

Se considera que la temperatura decrece 6.5°C por cada 1,000 m desde el nivel del mar hasta los 11,000 m de altura. La aceleración de la gravedad se considera constante para altitudes menores a 20,000 m.

B) Acelerómetros.

Utilizados para determinar el alabeo y cabeceo que se produzcan en la aeronave durante la campaña de detección.

Fabricante:	E.U.A.
Intervalo:	$\pm 2\text{g}$
Error:	$\pm 1\%$
Sensibilidad:	1019 mV / g
Máxima no linealidad:	1%

Operación: El sensor está formado por cuatro resistencias conectadas en puente y una masa móvil; dos de las resistencias disminuyen su magnitud cuando hay

una aceleración en un sentido debido a la fuerza que ejerce la masa móvil sobre ellas y las otras aumentan su valor, y viceversa cuando existe una aceleración en el sentido opuesto. El sensor se polariza con un voltaje CD. Se tienen 2.5 V de salida para aceleración cero. El sensor solo mide aceleración en un eje.

C) Sensor de Orientación.

Se utiliza una brújula electrónica para determinar el movimiento de guiñada de la aeronave durante el vuelo.

Fabricante:	E.U.A.
Intervalo de trabajo:	0 a 360 grados
Error:	± 0.5 grados RMS
Sensibilidad:	0.1 grado / 1.388 mV
Temperatura de operación:	0 a 50 °C

Operación: utiliza una bobina sensible al campo magnético terrestre, por lo que ofrece la orientación con respecto al Norte. La bobina ha sido suspendida en un fluido y por un arreglo mecánico para mantener la referencia aún en inclinaciones de ± 45 grados en dos ejes.

La bobina se alimenta con dos señales, una de referencia y la otra proveniente del campo magnético terrestre, la suma de éstas produce un flujo asimétrico en el núcleo. La señal resultante es procesada electrónicamente para dar como salida un voltaje de CD en el intervalo de 0.1 a 5.1 V de CD. La brújula cuenta con una bobina de Helmholtz para compensar las anomalías de campo magnético local. Este tipo de corrección no es afectada por el ángulo de inclinación como lo son todas las brújulas convencionales y electrónicas, es decir que el campo magnético lineal generado se compensa aún cuando el núcleo está inclinado.

D) Servomotores.

Utilizados para manipular los mecanismos de posicionamiento horizontal y de ajuste de distancia focal de la cámara móvil.

Fabricante:	Japón
Ángulo máximo de rotación	± 45 grados
Corriente	8 mA
Par	8 kg cm (6V de polarización)
Velocidad angular de operación	0.19 s / 60 grados (6V de polarización)

Operación: para controlar la velocidad de rotación de la flecha es necesario aplicar una onda cuadrada, con ciclo de trabajo variable y una amplitud igual a la del voltaje de polarización. Para una posición neutral la duración de los pulsos es de 1.5 milisegundos. El servomotor cuenta con un control electromecánico de posición automático, el cual permite que la flecha conserve la posición cuando se presente alguna perturbación en la carga.

3. ELECTRÓNICA DE CONTROL Y ADQUISICIÓN DE DATOS.

La utilidad de la instrumentación del contenedor estriba en la posibilidad de corregir el vector de apuntamiento de las cámaras para que las imágenes obtenidas se puedan armar en mosaicos y facilitar la interpretación de los resultados. La grabación de un número identificador en cada cuadro de video permitirá localizar cualquier imagen para facilitar su procesamiento.

3.1 Generalidades.

Para el diseño e implementación de las interfaces se consideraron los siguientes criterios: velocidad de respuesta adecuada, consumo de energía mínimo, transparencia para el resto del sistema y autonomía.

Las interfaces permiten controlar, mediante una computadora personal, lo siguiente:

- Las ocho funciones de operación de cada una de las cámaras de video, por medio del control remoto.
- La posición y distancia focal de una de las cámara mediante tres servomotores, dos de ellos rotan el servomecanismo en dos ejes y el tercero modifica la distancia focal.
- La codificación y registro de los datos que proporcionan todos los sensores descritos en el capítulo anterior.
- La grabación de un número identificador (etiqueta) en la pista de audio de las cintas de video de las cámaras. Este número permite asociar a una serie de imágenes los datos de los sensores y las coordenadas georreferenciadas que proporciona el receptor GPS.

- La sincronización en la grabación de los números de identificación de las tomas de las cámaras.
- La recuperación de las etiquetas mencionadas.
- La recepción y registro de la información proporcionada por el receptor GPS.
- La verificación constante de la operación adecuada del receptor GPS, de las cámaras de video y de los sensores disponibles en el sistema.

3.2 Decodificación de Direcciones.

Para establecer la comunicación entre la computadora y las interfaces, evitando conflictos con las otras tarjetas instaladas, es necesaria una etapa de decodificación de direcciones que activa o desactiva las interfaces por medio de un programa [5].

La decodificación de direcciones se realiza por medio de un comparador de dos números de ocho bits (SN74LS688), y dos decodificadores de tres a ocho líneas (SN74LS138); uno de ellos activa los dispositivos que reciben datos de la computadora (escritura en los registros SN74LS373) y el otro activa los dispositivos que envían datos (lectura de los bits que tienen los registros del convertidor analógico-digital AD547 y el ducto bidireccional SN74LS244).

Las interfaces se aíslan del ducto de datos mediante un controlador de tráfico bidireccional SN74LS245, que presenta alta impedancia cuando está inhabilitado. El bit AEN mantiene deshabilitadas las interfaces cuando hay acceso directo a memoria [5].

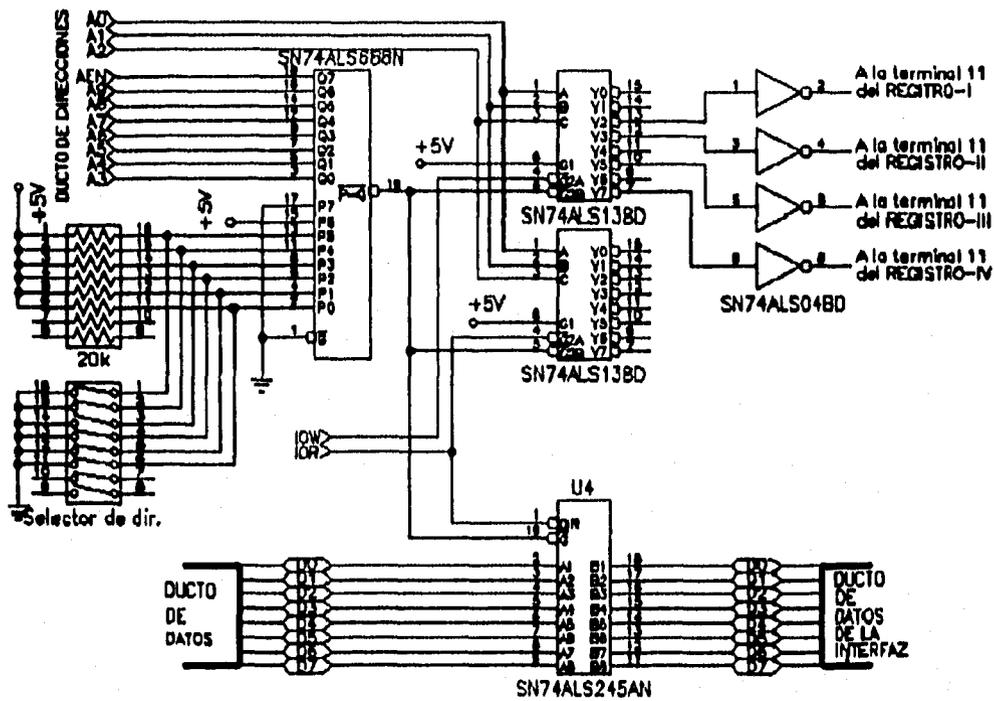


Figura 11. Diagrama del decodificador de direcciones.

3.3 Control de Funciones de las Cámaras.



Figura 12. Proceso de control de funciones de las cámaras

Para activar alguna de las funciones de las cámaras (grabar, reproducir, pausa, etc.), es necesario oprimir durante unos instantes el botón del control remoto correspondiente, su manejo desde la computadora se hace por medio de un registro de ocho bits, el cual tiene el dato asociado a la función que se desea activar y en otro registro se pone el número de la cámara al que se debe transmitir el comando.

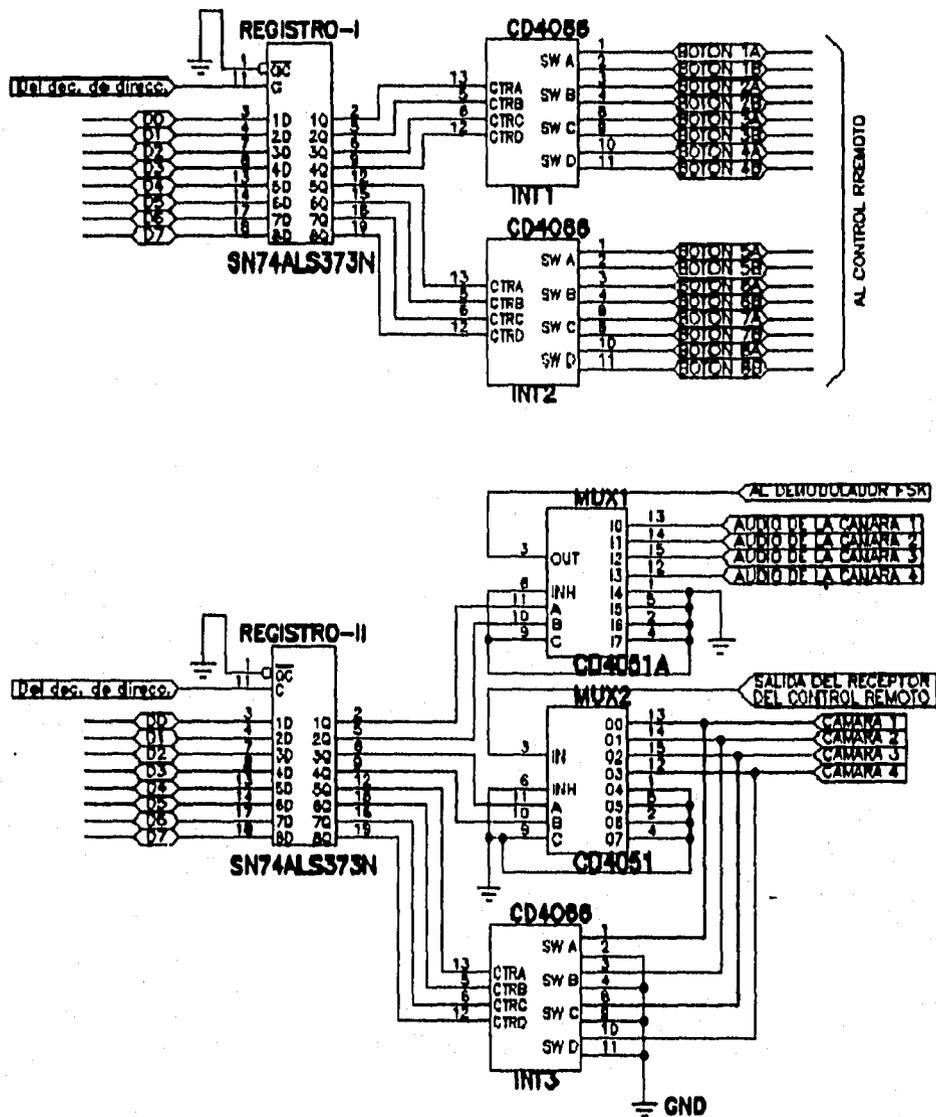


Figura 13. Diagrama de control de funciones de cámaras

Dependiendo de la función que se seleccione, la computadora transmite el dato correspondiente al registro I, sus bits de salida proporcionan las señales de control para cerrar uno de los ocho interruptores electrónicos CD4066; el que está

conectado en paralelo con el botón del control remoto de la función seleccionada y mantiene abiertos los otros siete interruptores. El enlace entre el control remoto y su receptor es óptico, sin embargo para esta aplicación éste enlace se sustituyó por cables; como la señal que transmite el control remoto está deformada, se pasa por dos inversores para enviar al receptor del control remoto una señal cuadrada sin distorsión.

Los bits del registro II controlan las líneas de selección del multiplexor analógico CD4051, con el que se distribuye la señal del receptor a las cuatro cámaras de video, lo que permiten usar un solo dispositivo de control remoto (transmisor/receptor) para todas las cámaras.

Para ahorrar energía las cámaras utilizadas se apagan después de cinco minutos de no ejecutarse alguna función, por lo que es necesario reactivarlas conectando momentáneamente las entradas del receptor del control remoto a tierra mediante los interruptores electrónicos CD4066.

3.4 Control del mecanismo de posicionamiento y ajuste de distancia focal.



Figura 14. Proceso de control del mecanismo de posicionamiento de cámara y ajuste de distancia focal.

Los pulsos de control para los servomotores se generan con moduladores de ancho de pulso (PWM), están formados por dos temporizadores LM555, donde uno funciona como astable y el otro como monoestable. La duración del pulso a la salida del monoestable depende tanto del voltaje en las terminales 7 y 6 como del voltaje de control, terminal 5, ver diagrama de la figura 15 [9].

Los pulsos de salida de los moduladores se transmiten a los servomotores a través de un optoacoplador 4N35, este circuito permite aislar a la computadora de los servomotores.

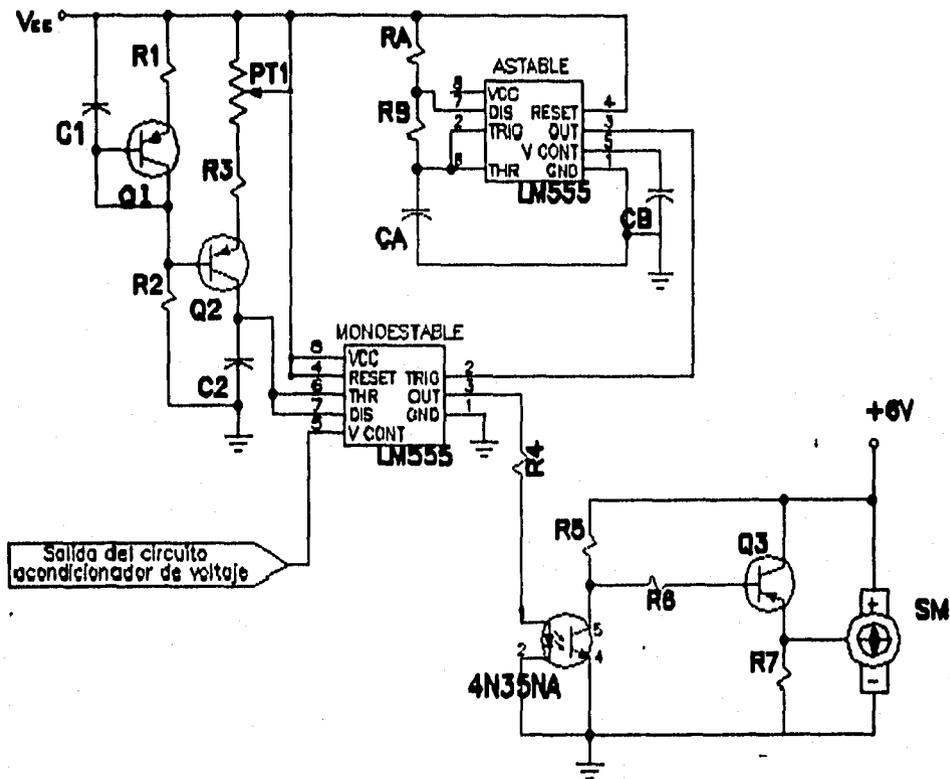


Figura 15. Diagrama del modulador por ancho de pulso (PWM).

Para la duración de los pulsos del circuito de la figura anterior se tiene la siguiente expresión:

$$t = C2 \frac{1}{2} V_{CONT} \left[\frac{(R1 + R2)(PT1 + R3)}{R1(V_{CC} - V_{EB})} \right]$$

La duración del pulso varía linealmente con respecto al voltaje de control, consecuentemente la rotación del servomotor también depende linealmente de este voltaje.

Para controlar a los moduladores desde la computadora, se escribe un dato en el registro, el cual se convierte a un voltaje analógico por medio del circuito integrado DAC0800, el voltaje de salida se ajusta al intervalo de 6 a 10.6 V. Este circuito acondicionador está formado por tres amplificadores operacionales: un seguidor, un sumador-inversor y un inversor. El voltaje de salida del circuito es el voltaje de control que necesitan los monoestables para que la duración de los pulsos que entregan sea de un milisegundo, como mínimo, y 2 milisegundos como máximo. El voltaje de salida del circuito acondicionador se distribuye a los monoestables por medio del multicanalizador CD4051 y los muestreadores-retén [10].

El multicanalizador se incluye para emplear solo un convertidor digital-analógico, sus líneas de selección son controladas por dos bits del registro; los muestreadores-retén se utilizan para mantener el voltaje que se aplica a los monoestables durante varios segundos, evitando tener que renovarlo constantemente; el estado de los LF398 (muestra o retención) se controla por otros tres bits del mismo registro.

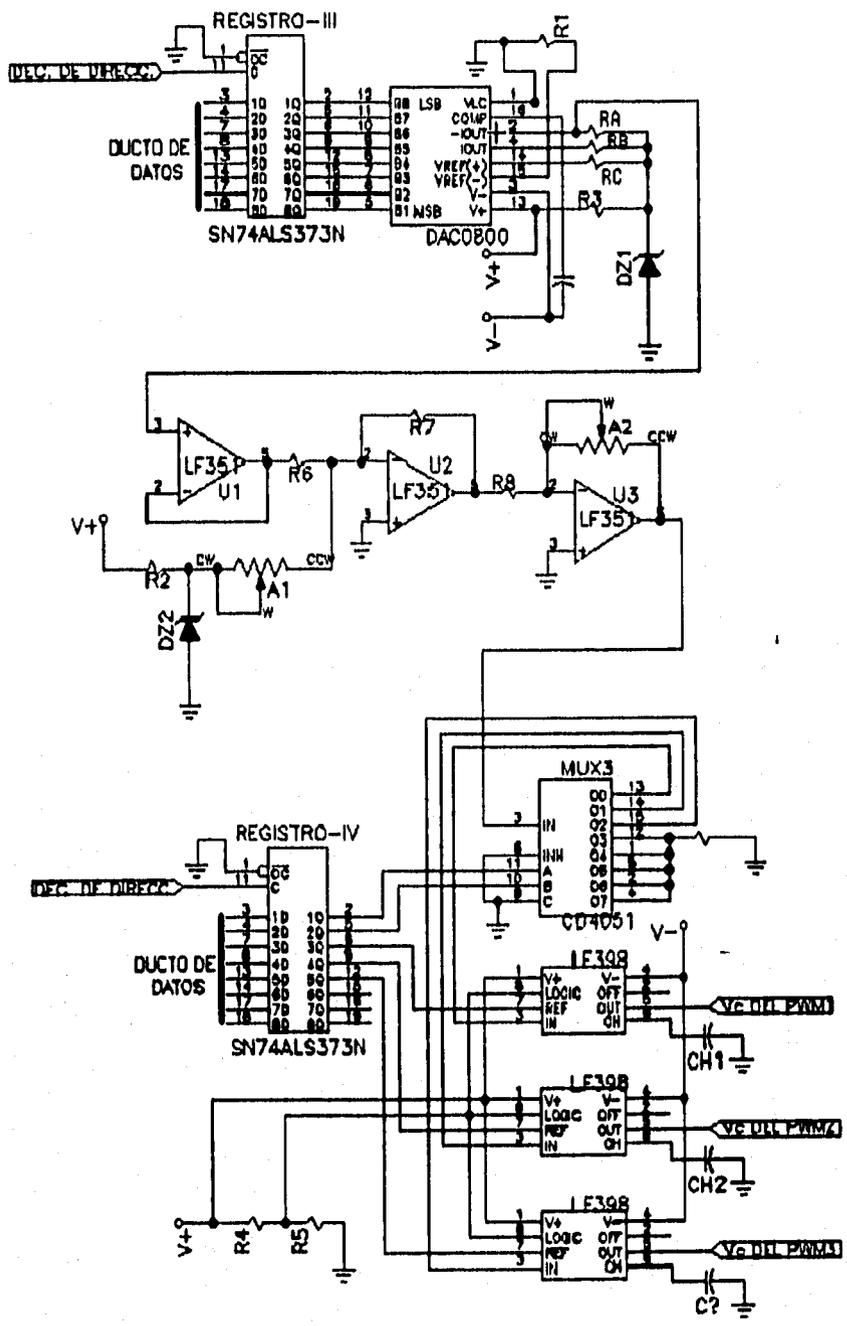


Figura 16. Diagrama de conversión Digital-Analógica, multicanalización y muestreadores-retén.

3.5 Adquisición de datos de sensores.



Figura 17. Proceso de adquisición de datos de los sensores.

A los sensores se les adicionó una etapa de amplificación con el fin de que proporcionen voltajes analógicos en el intervalo de 1 a 9 V. [8]. Para almacenar estos datos en la computadora se utiliza un convertidor analógico-digital de 12 bits AD547AK y un multicanalizador de 16 canales a una línea, que permite registrar los datos de hasta 15 sensores. Un muestreador-retén mantiene estable el voltaje en la entrada del convertidor el tiempo suficiente para que se realice la conversión, que tarda como máximo 35 microsegundos [11].

Para registrar el dato en la computadora de alguno de los sensores, primero se selecciona el sensor transmitiendo un número de cuatro bits al registro que controla las líneas de selección del multiplexor analógico ADG506 y se cambia la operación del circuito integrado LF398 de muestra a retención; cambiando el primer bit del registro de cero a uno. La computadora manda la señal de inicio de conversión cambiando el estado de las terminales 4, 5 y 6 del AD547AK mediante el decodificador de direcciones, al terminar la conversión, una palabra de 12 bits queda disponible en los registros internos del convertidor y se almacena en la computadora.

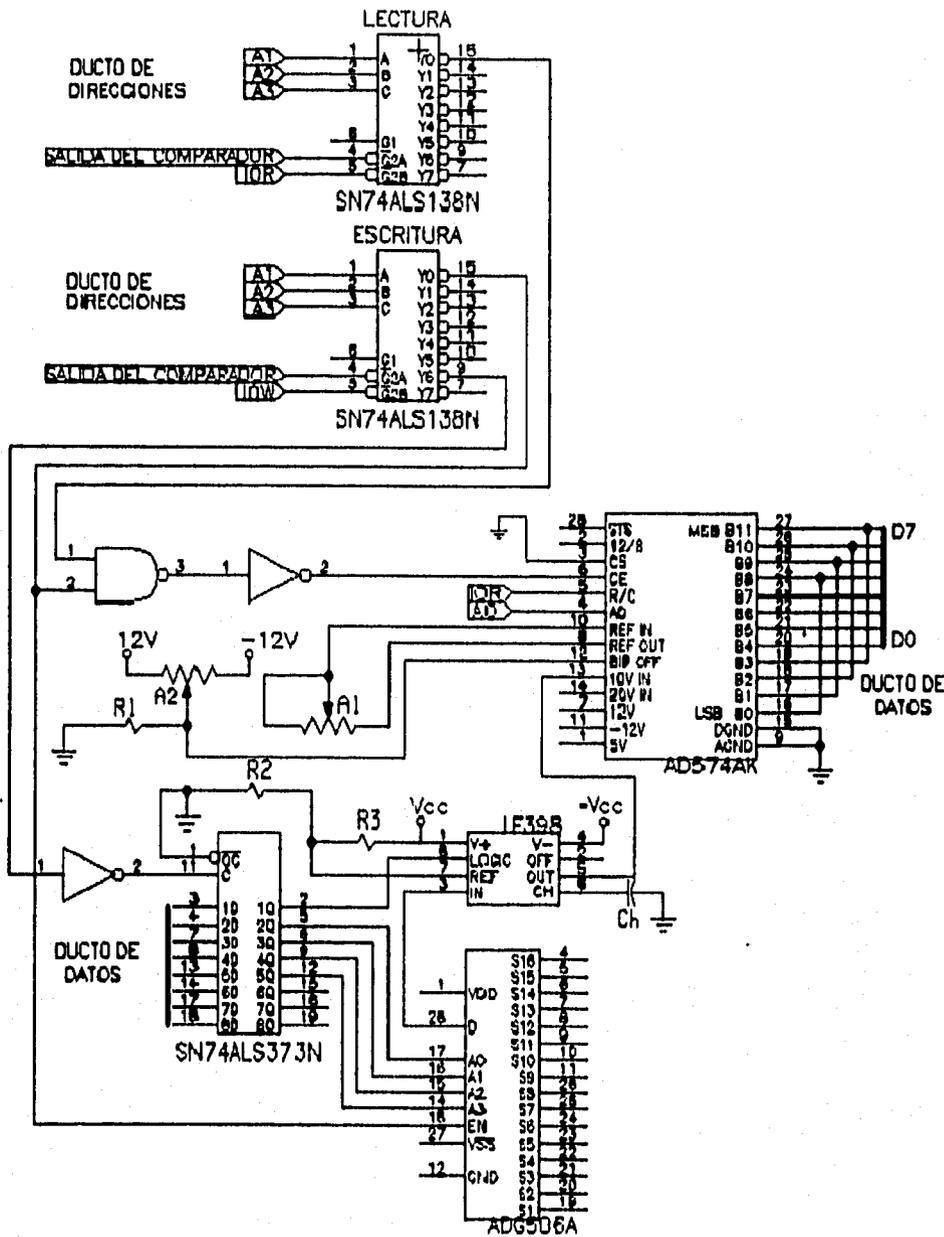


Figura 18. Diagrama de conversión Analógica-Digital.

Durante la operación del sistema, es importante conocer si las cámaras están grabando o no. Para indicárselo a la computadora se aprovecha que la cámara

tiene un LED que se ilumina cuando ésta se encuentra grabando. Para transmitir el estado del LED a la computadora, se emplea una fotoresistencia y un comparador, el cual tiene como voltajes de salida: 0 ó 5 V. DC, cuando el LED está encendido el voltaje de salida del comparador es de 5 V ("1" lógico), éste estado se transmite al ducto de datos de la computadora por medio de un ducto bidireccional SN74ALS244, dependiendo del estado de estos bits se conoce cuales cámaras están grabando [5].

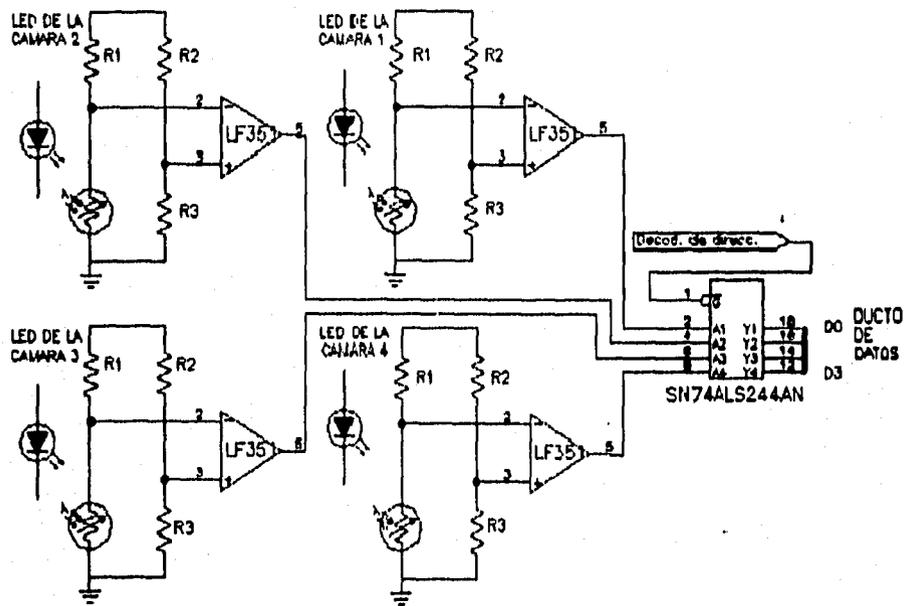


Figura 19. Diagrama de verificación de grabación de cámaras.

3.6 Etiquetación de cada imagen.

La etiqueta de cada imagen se graba en la pista de audio de las cintas de video, a través del micrófono de cada cámara. La máxima amplitud de la señal que se le puede aplicar al micrófono es de 50 mV (voltaje de saturación), y su respuesta en frecuencia se muestra en el apéndice B, ahí puede apreciarse que el canal de

audio de la cámara y la cinta en conjunto actúan como un filtro paso bajas con una frecuencia de corte de alrededor de los 10 kHz.

A) Grabación



Figura 20. Proceso de grabación de etiquetas de imágenes.

La información para cada imagen se transmite por el puerto serie de la microcomputadora, esta señal es una secuencia de unos y ceros en formato RS-232 (-12V para un uno lógico y +12V para un cero lógico), con componentes de alta frecuencia que están fuera del ancho de banda que puede registrar el canal de audio. Por esta razón se diseñó un circuito de codificación por corrimiento en frecuencia (FSK, Frequency Shift Keying), así la señal binaria es transformada en una senoide con una amplitud fija de 45 mV y dos frecuencias (6.1 kHz define un uno lógico y 8.5 kHz define un cero lógico). Las señales FSK presentan alta inmunidad al ruido, y ya que el sistema operará en un ambiente de alta interferencia electromagnética, éstas resultan adecuadas; además de operar en el intervalo del canal de audio de las cámaras.

La transmisión se realiza a 4800 bits por segundo, posibilitando grabar hasta 13.2 bytes de información en cada imagen. Se estableció el siguiente protocolo: un byte de inicio, ocho bytes de información y 4.2 bytes no se transmiten y proporcionan tiempo para la recuperación de información.

La secuencia de unos y ceros contiene la información de las coordenadas del centro de toma de las cámaras proporcionadas por el receptor GPS

B) Recuperación



Figura 21. Proceso de recuperación de la etiqueta de cada imagen.

Para obtener las etiquetas, se hace una demodulación coherente mediante una malla de fase encadenada (PLL), el multicanalizador permite usar un PLL para las 4 cámaras de video, la señal que entrega el PLL es acondicionada para que se pueda transmitir a la microcomputadora a través de su puerto serie.

Para poder demodular una señal FSK, sus frecuencias de codificación deben encontrarse dentro de las bandas de captura y de amarre. La función del detector de fase es multiplicar la señal de entrada con la señal del oscilador controlado por voltaje (VCO).

Cuando el VCO del PLL está oscilando a la misma velocidad angular de la señal de entrada ($\omega_1 t = \omega_0 t$) y la diferencia entre ambas señales es su ángulo de fase ($\theta_1(t)$, $\theta_2(t)$), se dice que el PLL está amarrado con la señal de entrada. Suponiendo que la señal de entrada (e_1) con una amplitud E_1 y la señal del VCO (e_0) con una amplitud E_0 , son senoidales, entonces:

$$e_1 = (2)^{1/2} E_1 [\text{sen}(\omega_0 t + \theta_1(t))]$$

$$e_0 = (2)^{1/2} E_0 [\text{sen}(\omega_0 t + \theta_2(t))]$$

la señal de salida del comparador de fase ($e_1 e_0$) es:

$$e_1 e_0 = 2 E_1 E_0 \text{sen} [\omega_0 t + \theta_1(t)] \text{Cos} [\omega_0 t + \theta_2(t)]$$

$$e_i e_o = E_i E_o \sin [\theta_1(t) - \theta_2(t)] + \sin [2\omega_0 t + \theta_1(t) + \theta_2(t)]$$

con el filtro pasobajas se elimina el segundo término del lado derecho de la igualdad, quedando la señal e_r :

$$e_r = E_i E_o \sin [\theta_1(t) - \theta_2(t)]$$

la anterior expresión representa un voltaje que depende de la diferencia entre los ángulos de fase de las señales de operación.

En esta aplicación se utiliza un circuito para comparar el voltaje a la salida del filtro pasa bajas con un voltaje de referencia (terminal 6 del LM565) y así poder generar la señal correspondiente al formato RS-232.

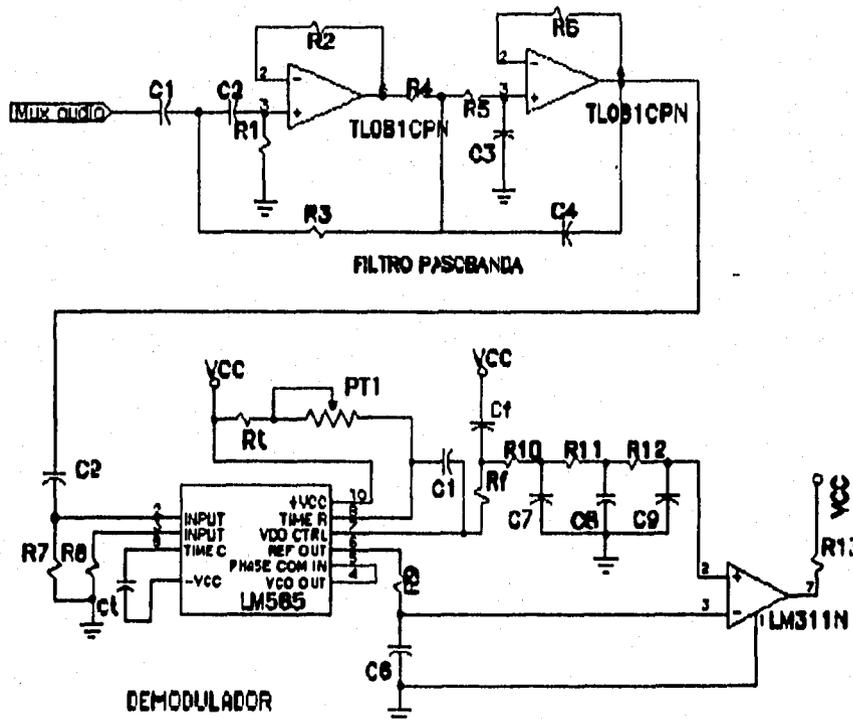


Figura 22. Diagrama de la configuración de PLL utilizada.

4. PROGRAMA DE CONTROL.

4.1 Generalidades.

El sistema de adquisición de imágenes en su conjunto es controlado por un programa en lenguaje C a través de una microcomputadora.

Dicho programa fue desarrollado considerando los siguientes criterios: rapidez de respuesta, optimación de recursos disponibles, transparencia al usuario y flexibilidad para su expansión.

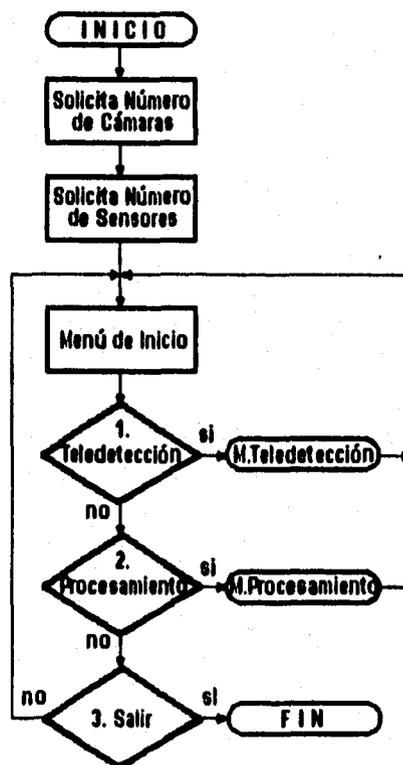


Figura 23. Diagrama de flujo general del sistema.

Las tareas principales de este programa son:

- grabar la posición X, Y, Z de la aeronave a partir de la información obtenida del receptor GPS.
- Grabar un número identificador para cada cuadro de imagen.
- Recuperar el número identificador de cada imagen.
- Almacenar los datos de los sensores.
- Comprobar que las cámaras estén grabando durante el vuelo.

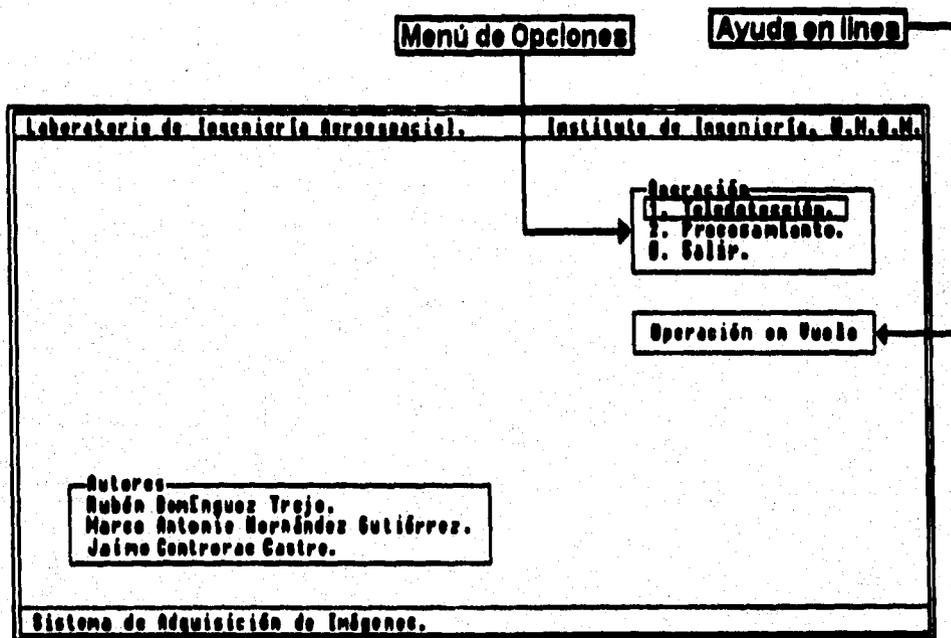


Figura 24. Pantalla inicial del programa de control y supervisión del sistema.

El sistema tiene dos modos fundamentales de operación:

- En vuelo, para realizar la campaña de teledetección,
- En tierra, para el análisis de los datos e imágenes obtenidos.

Al iniciar su funcionamiento, el programa solicita al usuario el número de cámaras y el número de sensores que operarán en el sistema. Posteriormente se despliega una pantalla inicial que permite elegir uno de los dos modos de control del sistema, además de una pequeña ayuda en línea.

En cada modo de operación se despliega en pantalla información sobre sus diferentes elementos activos, lo que permite verificar su funcionamiento adecuado y actuar en consecuencia. Además, se tiene una pequeña ayuda en línea sobre las acciones de las funciones permisibles en cada modo de operación. Cada modo de operación requiere de ciertos datos para inicializar su operación, los cuales son solicitados al usuario, antes de activarse.

Cuando algún elemento del sistema no opera adecuadamente, el desperfecto es notificado al usuario. Si la falla es determinante de la función de que se trate, el programa abortará, ya que su continuidad proporcionaría información no confiable.

4.2 Modo de operación en Vuelo.

Este modo de operación fue desarrollado para cubrir la etapa de teledetección, durante la cual se requiere recibir y almacenar la información que envía el receptor GPS y grabar una etiqueta de identificación en cada imagen tomada por las cámaras de video. Este proceso de identificación se lleva a cabo durante el regreso del barrido vertical, tomando la sincronía de una de las cámaras. Como no es posible sincronizar todas las cámaras para que el inicio de cada imagen sea el mismo, entonces en este proceso en particular se puede estar introduciendo un error hasta de un cuadro en la grabación del número en las cámaras restantes.

Estas tareas se realizan a través de rutinas de interrupción definidas en el programa, con el fin de que el controlador de interrupciones de la microcomputadora se encargue de las mismas, optimando la ejecución del resto del programa.

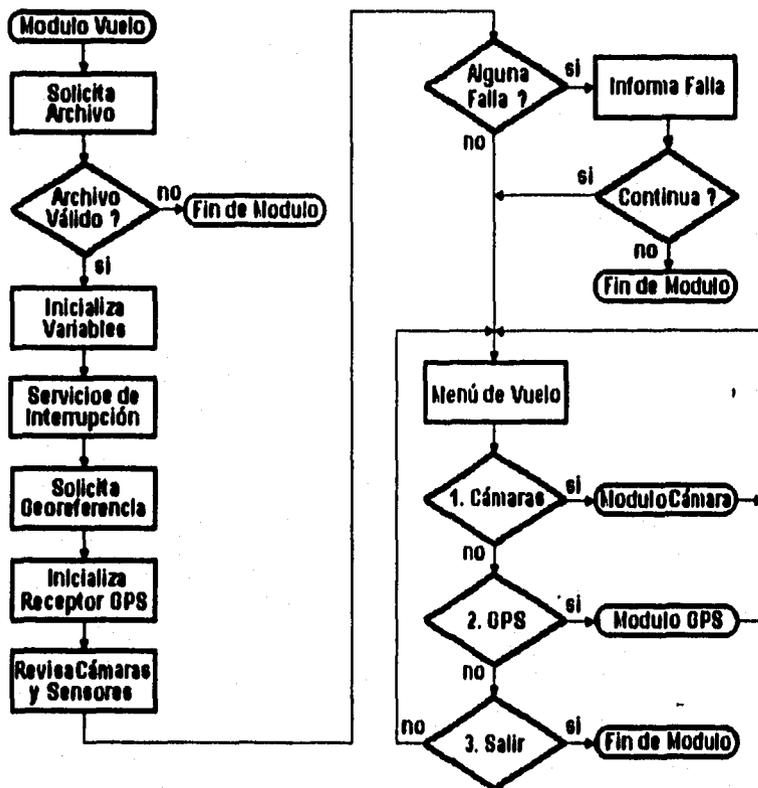


Figura 25. Diagrama de flujo del Modulo de Vuelo.

Se realizan las siguientes operaciones:

- Abre el archivo para almacenar los datos proporcionados por el receptor GPS y los sensores.
- Activa al receptor GPS, solicitando la posición geográfica inicial, que es esencial para el sistema.

- Comprueba el funcionamiento de los sensores, de las cámaras de video, del mecanismo de posicionamiento y el ajuste de distancia focal de la cámara móvil.

Una vez proporcionado el nombre del archivo de datos, el modo de operación queda activo. La omisión de nombrar el archivo de datos impedirá seguir adelante al programa, que retorna a la pantalla inicial.

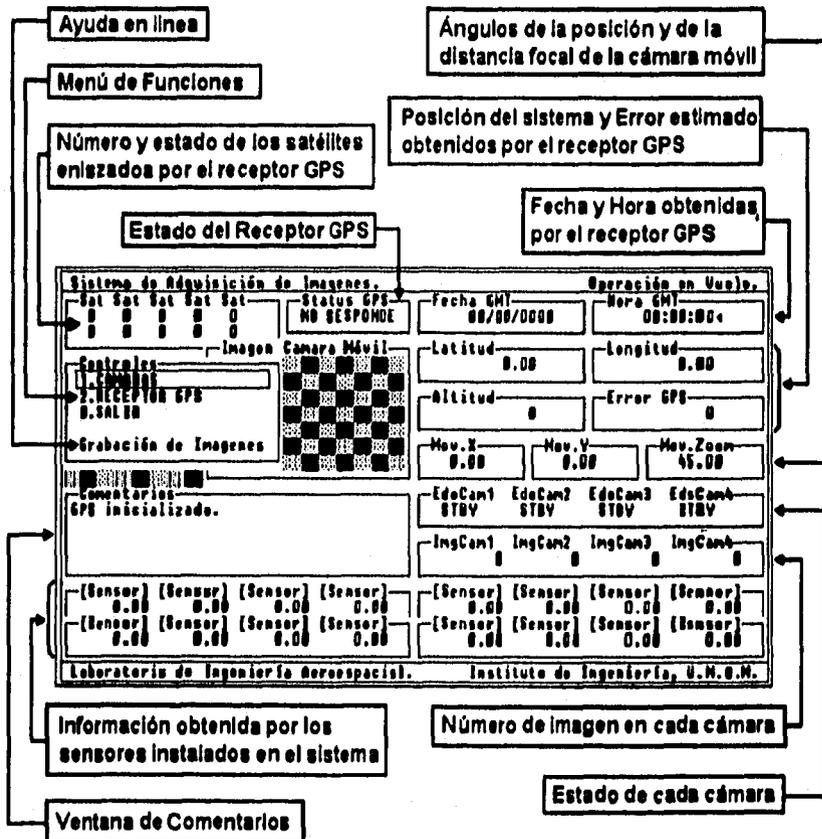


Figura 26. Descripción de la pantalla del modo de operación de Vuelo.

La información en pantalla es actualizada constantemente mientras el modo de operación se encuentra activo.

Las funciones controladas en este modo de operación son:

- manejo del receptor GPS,
- control de las cámaras,
- manipulación de la posición y distancia focal de la cámara móvil.

4.2.1 Manejo del receptor GPS.

Comprende las funciones relacionadas al manejo y operación directa del receptor GPS, indicando los requerimientos y avisos de cada función en la ventana de comentarios.

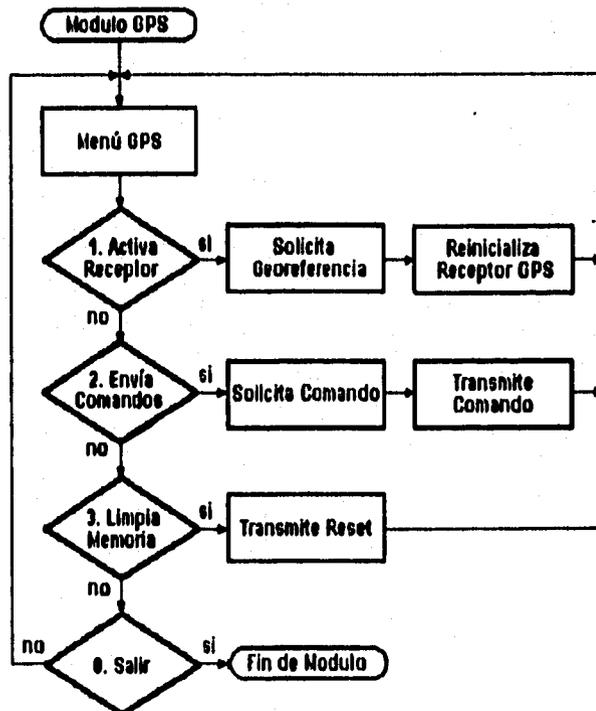


Figura 27. Diagrama de flujo del Modulo de GPS en la operación de Vuelo.

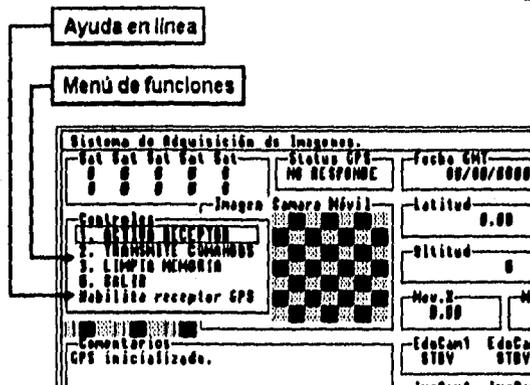


Figura 28. Funciones de control del receptor GPS.

Activación del Receptor. Habilita la operación del receptor GPS, para lo cual se requiere proporcionar la posición geográfica del sistema (altitud, longitud, latitud y datum). La información solicitada tiene un formato definido por lo que se presenta un ejemplo entre corchetes, no obstante, se puede optar por tomar los datos por asignación que el sistema incluye. Estos establecen la posición geográfica del edificio 12 del Instituto de Ingeniería, en Ciudad Universitaria, donde se localiza el Laboratorio de Ingeniería Aeroespacial.

El sistema intenta posicionarse respecto a los datos proporcionados por el usuario, lo que toma sólo algunos segundos.

Si se desconoce la posición geográfica exacta del sistema, como sucede cuando se realiza una campaña de teledetección sobre una nueva región, el usuario debe dar ceros en los datos que se solicitan. Así, el receptor procede a autolocalizarse, lo que lleva algunos minutos. Cabe aclarar que, si el sistema es alimentado con una posición geográfica errónea, el receptor GPS corregirá dicha posición, condición que llevaría más tiempo al receptor para habilitarse, ya que primero

intenta posicionarse respecto a los datos proporcionados, después los desecha y finalmente se autolocaliza.

Además de lo anterior, se envían las ordenes correspondientes para que el receptor proporcione la siguiente información cada segundo: latitud, longitud, altitud, fecha, hora, velocidad, orientación, número de los satélites enlazados y su estado de comunicación y el estado del receptor. Estos datos son esenciales para realizar el posterior procesamiento de las imágenes adquiridas, así como para conocer la posición del sistema mientras se realiza la campaña de teledetección.

Envío de comandos. Opción mediante la cual se envían comandos específicos al receptor GPS de manera manual. Para el uso de esta opción se requiere tener un conocimiento profundo de los comandos de operación del receptor GPS, ya que estos cuentan con una estructura definida para ser interpretados. Un comando enviado con una estructura incorrecta, es desechado por el receptor, es decir, se considera como no-enviado. No obstante, dado el diseño del programa, no se contempla el uso regular de esta función, ya que la información que se requiere enviar y recibir del receptor GPS se define en la función de "Activación del Receptor", evitando así, que el usuario opere manualmente el receptor.

Borra memoria. Aún cuando el sistema esta diseñado para funcionar en un ambiente severo, es posible que el receptor se "congele", es decir, que no envía información. Esta función limpia la memoria del receptor, permitiendo salir del estado inactivo, manteniendo vigente la información de posición geográfica y transmisión de datos, lo que permite continuar la campaña de teledetección del sistema. Si una vez activada esta función el receptor no responde, será necesario reactivar el sistema nuevamente. Para conocer si el sistema responde o no, se debe observar si en la ventana de "Status GPS" o en la ventana de "Hora GMT", hay cambio.

4.2.2 Manejo de Cámaras.

Comprende funciones de teledetección y control durante el vuelo. Es importante mencionar que durante la grabación de imágenes, se revisa constantemente el adecuado funcionamiento de las cámaras, de los sensores, del receptor GPS y de los servomotores del mecanismo móvil del sistema, informando al usuario de cualquier falla que pudiera presentarse.

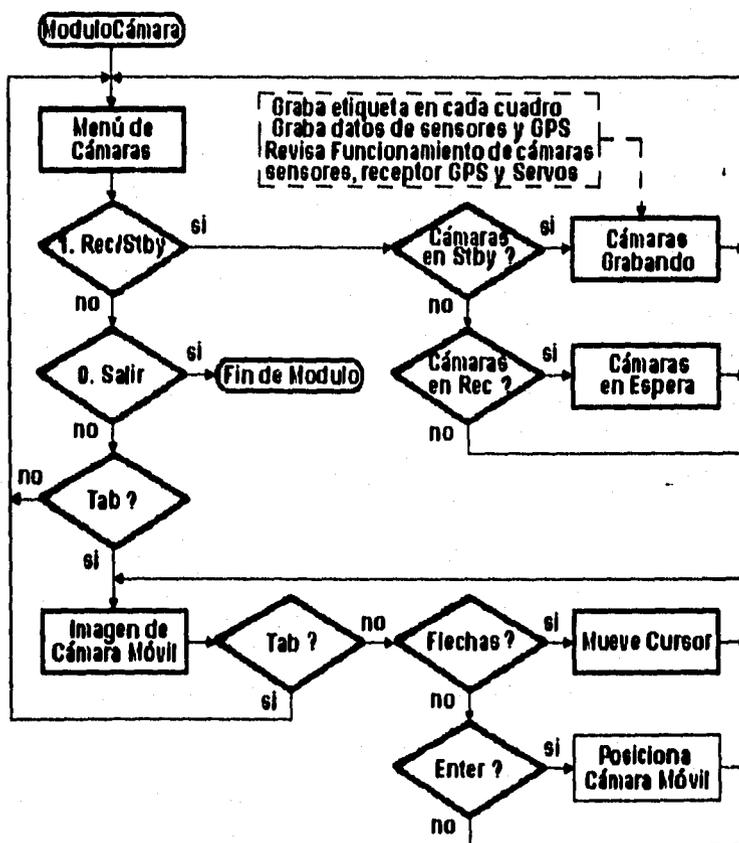


Figura 29. Diagrama de flujo del Modulo de Detección operando en vuelo.

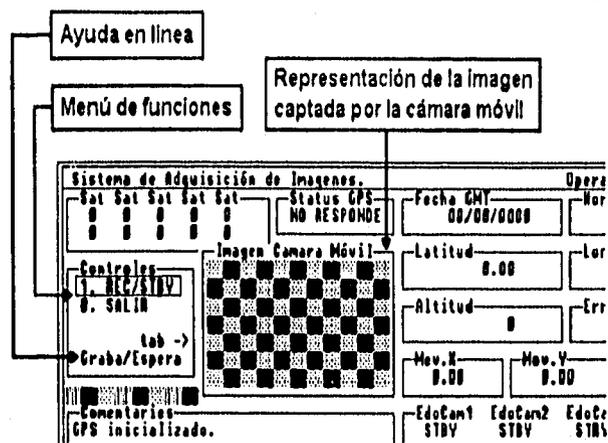


Figura 30. Funciones del control de cámaras y de la orientación y de la distancia focal de la cámara móvil.

Record/Standby. Esta función permite controlar las cámaras en su operación de grabación y espera, durante la adquisición de imágenes. Al accionar esta función, si el estado de las cámaras es STBY, su operación cambiará a REC; y viceversa. En este modo de operación es fundamental conocer cuanto antes el estado real de las cámaras, ya que si las cámaras no graban, no se está cumpliendo con la función básica del sistema: la adquisición de imágenes. Por esta razón, se tiene una fotoresistencia que verifica constantemente que cada cámara esté grabando. Cuando una cámara esta en el modo de grabación, pero realmente no está grabando, se informa al usuario del hecho y se indica en la ventana correspondiente su estado. En caso de que todas las cámaras no estén grabando, entonces se informa al usuario y termina el programa.

Durante la toma de imágenes, cada cuadro es etiquetado con un número único, que es grabado en la pista de audio de la cinta de video de cada cámara, que se transmite a través de la entrada de micrófono de la cámara, por medio del puerto serie de la computadora. Este número está asociado a la información del centro

de toma proporcionada por los sensores y el receptor GPS y almacenados en el disco duro de la computadora.

Cabe mencionar que las cámaras pueden grabar, aún cuando el receptor GPS no esté geoposicionado, lo cual no se recomienda, ya que los datos del centro de toma serían incorrectos. Por esta razón, se recomienda poner a grabar a las cámaras de video, una vez que el estado del receptor GPS esté habilitado. Ya que sólo cuando las cámaras graban, cada cuadro es etiquetado, y su número único asociado es almacenado en disco duro junto a la información proporcionada por los sensores y por el receptor GPS.

Imagen Cámara Móvil. En este caso, se tiene una ventana que representa el cuadro de imagen captado por la cámara móvil. Dentro de esta rutina es posible indicar hacia donde se quiere apuntar la cámara móvil. En esta ventana se puede navegar mediante las flechas del teclado de la computadora, de esta forma se puede elegir una área de interés, observando las imágenes en otro monitor.

Para elegir un área de interés en la ventana "Imagen Cámara Móvil", debe ubicarse la posición y presionar ENTER. En ese momento el mecanismo de posicionamiento apuntará la cámara hacia esa zona y se realizará un acercamiento máximo con el mecanismo de ajuste de la distancia focal. Considerando la velocidad, orientación y altitud de la aeronave, la zona elegida será seguida por el mecanismo de posicionamiento mientras se encuentre en el campo visual accesible a la cámara.

Una vez que la zona elegida sale del campo visual de la cámara, el mecanismo de posicionamiento coloca la cámara verticalmente, y el mecanismo de control de distancia focal realiza un alejamiento total. En este momento se está en condiciones de elegir otra área de interés.

La posición de los motores que controlan los mecanismo de apuntamiento y ajuste de distancia focal de la cámara móvil, se muestran en las ventanas de la pantalla

tituladas: "Mov.X", grados de movimiento en el eje X del plano; "Mov.Y", grados de movimiento en el eje Y del plano; "Mov.Zoom", estado de la distancia focal de la cámara móvil. Esta información se muestra para que el usuario observe la orientación en que se encuentra la cámara móvil, respecto a la vertical.

4.4 Modo de operación en Tierra.

Este modo de operación se proyectó para procesar los datos e imágenes obtenidas durante una campaña de teledetección. En dicha tarea se lee la etiqueta de identificación de cada imagen de las cámaras de video, para que sean desplegadas de manera simultánea en varios monitores y sea posible comparar imágenes obtenidas al mismo tiempo. Esto se logra mediante rutinas de interrupción declaradas en el programa; así, el controlador de interrupciones de la microcomputadora se encarga de atender esta función, dejando al resto del programa los demás procesos del sistema.

Este modulo realiza las siguientes operaciones:

- solicita y verifica la existencia del archivo de trabajo,
- abre el archivo de trabajo,
- revisa el funcionamiento de las cámaras de video.

Una vez realizado lo anterior, el modo de operación está activado. Si el nombre del archivo de trabajo no es proporcionado, el programa regresa a la pantalla inicial, dado que es del archivo de donde se obtiene la información inicial para llevar a cabo el procesamiento posterior.

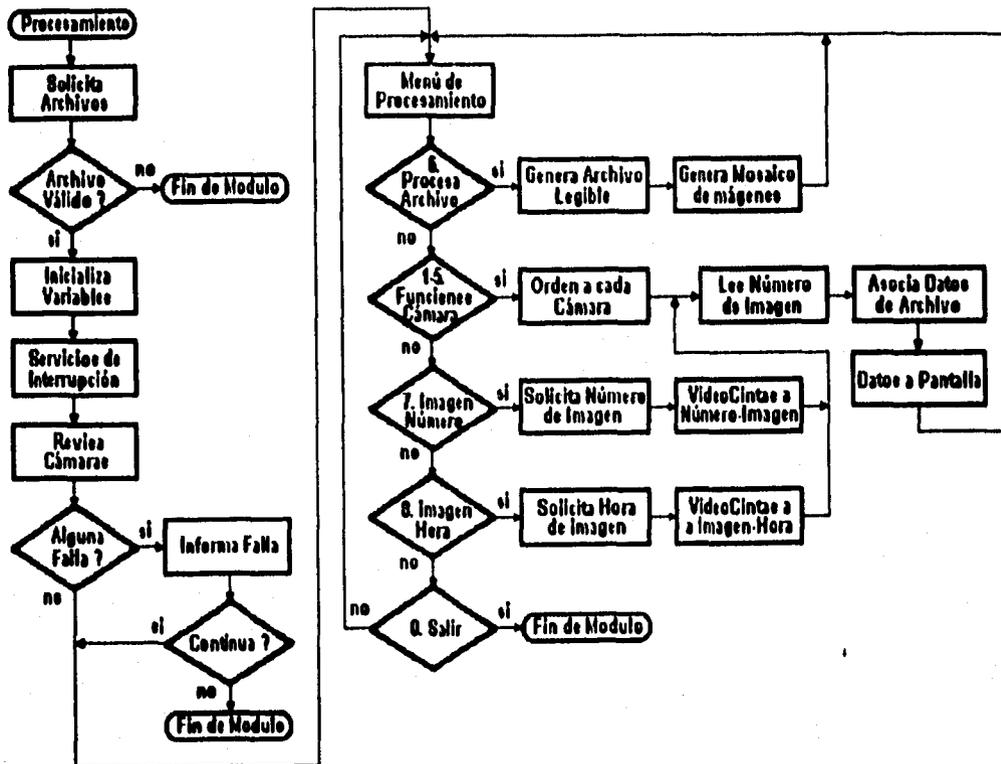


Figura 31. Diagrama de flujo del Modulo de Tierra.

Todos los datos de pantalla son obtenidos del archivo de trabajo, los cuales son actualizados cada vez que cambia el número de imagen leído de las cámaras de video. Igualmente, todas las cámaras se sincronizan nuevamente al pasar a otro número de imagen.

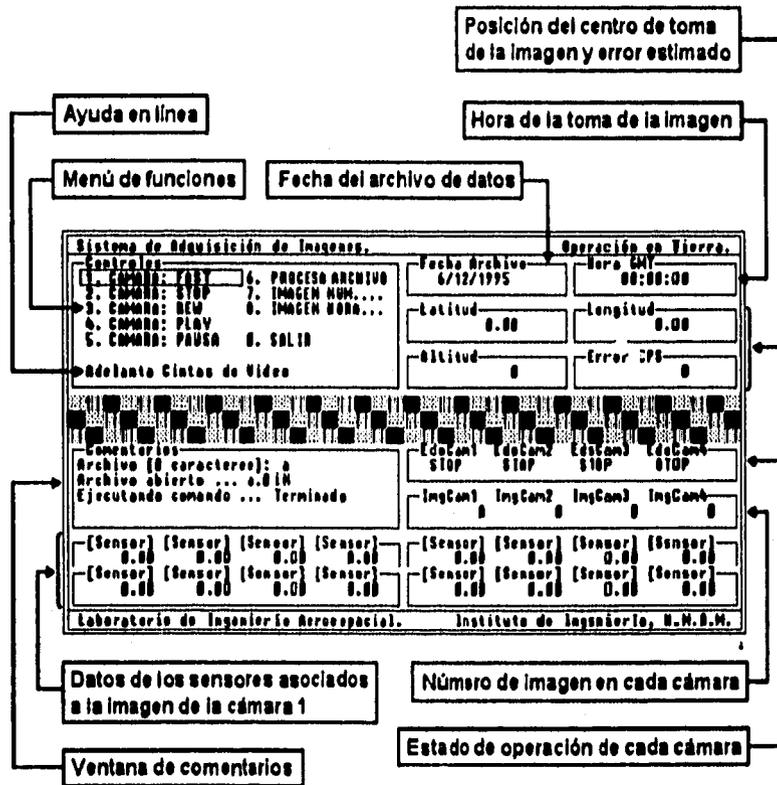


Figura 32. Descripción de la pantalla del modo de operación de Tierra.

FF, Stop, Rew, Play, Pause. Controla las funciones inherentes de las cámaras de video en su fase de digitación de imágenes para su corrección geométrica, procesamiento y conformado de mosaicos, como son: adelantar, detener, regresar, reproducir y pausa. Dichas ordenes son enviadas secuencialmente a las cámaras del sistema por medio del control remoto común. Al tiempo de reproducir las imágenes, se lee el número de identificación de cada una en la pista de audio de las cintas de video y se extraen los datos asociados a dicho número del archivo de trabajo.

Cada vez que una de estas funciones cesa, cambia el estado de operación de las cámaras del sistema y las imágenes de las cámaras son sincronizadas de forma

automática. Es importante señalar que esta sincronización se lleva a cabo sólo si las cámaras son manipuladas mediante el programa, por lo que un manejo manual de las cámaras provocaría inevitablemente un defasamiento en las imágenes presentadas por las cámaras, lo cual constituiría un serio problema para la formación de mosaicos y la interpretación física de las imágenes.

Procesa Archivo. Genera un archivo de tipo texto a partir del archivo de trabajo generado durante la campaña de percepción remota, a la vez que complementa la información proporcionada por el receptor GPS, ya que las imágenes son tomadas a una frecuencia de 30 cuadros por segundo y el receptor GPS proporciona la información solicitada cada segundo. La correlación de estos datos se realiza por interpolación lineal.

Posteriormente crea un arreglo matricial de imágenes de la zona sobrevolada durante la campaña, generándose un archivo con los números de cuadro que cubre la zona, con un traslape entre imágenes generalmente del 60%.

Imagen Num... Ductoca el número de imagen indicado por el usuario, en todas las cámaras de video del sistema y las despliega simultáneamente. Posteriormente, ductoca los datos asociados al número en el archivo de trabajo y los despliega en pantalla. Estos datos son: Georreferencia del centro de toma (Altitud, Latitud, Longitud y Error estimado), hora de toma (Hora GMT), orientación del centro de toma (Sensores: Alabeo, Cabeceo, Rumbo).

Imagen Hora... Ductoca una imagen según su hora de toma en el archivo de trabajo, una vez localizada y de acuerdo con el número de imagen, las cámaras son sincronizadas. Posteriormente, se ductocan los datos asociados a dicho número en el archivo de trabajo y se despliegan en pantalla. Estos datos son: Georreferencia del centro de toma (Altitud, Latitud, Longitud y Error estimado),

hora de toma (Hora GMT) y orientación del vector de apuntamiento (Sensores: Alabeo, Cabeceo, Rumbo).

5. PRUEBAS EXPERIMENTALES Y DISCUSIÓN DE RESULTADOS.

Para probar las interfaces desarrolladas se crearon programas de control y una vez verificado su funcionamiento se integraron al programa general de control y supervisión.

En todas las interfaces se comprobó su compatibilidad con el funcionamiento del resto del sistema; la computadora, las otras interfaces, y los dispositivos de control y comunicación (v.g: receptor GPS, cámaras, sensores, servomotores).

5.1 Pruebas al control de funciones de las cámaras de video.

Para probar el funcionamiento de esta etapa se utilizó el circuito acondicionador de la señal de control remoto para ser transmitida por cable, una cámara, un control remoto y un osciloscopio. Con el osciloscopio se compararon en amplitud y frecuencia las señales de salida del control remoto y de salida del circuito acondicionador, además se verificó que la señal enviada por el receptor de control remoto fuera distribuida a las cuatro salidas del multiplexor analógico. Posteriormente se transmitieron mediante el programa cada una de las ocho funciones a las cámaras, observándose un funcionamiento correcto. Esta prueba fue para establecer los tiempos de respuesta de las cámaras en el programa antes de poder ejecutar otra orden.

5.2 Pruebas de etiquetación de imágenes.

Se evaluó la respuesta del circuito para grabar y leer información en la pista de audio de la cinta de video, utilizando un generador de funciones, una cámara, un osciloscopio, el circuito modulador de FSK y el circuito demodulador de FSK.

Para grabar información, el generador de funciones alimenta al circuito modulador de FSK con una señal cuadrada de 2400 Hz y con niveles de voltaje según el

formato RS-232. La señal tiene una frecuencia de 2,400 Hz. ya que para una velocidad de 4,800 bauds es el tiempo que dura un bit es de 0.21 ms, que es el tiempo que dura medio ciclo de la señal proporcionada por el generador. La salida del modulador se alimenta al conector del micrófono de la cámara para lograr así la etiquetación de las imágenes. Durante el período de grabación se observaron las señales del generador de funciones y del modulador FSK, verificando que frecuencias y amplitudes se mantuvieran dentro del intervalo de valores adecuados.

Posteriormente, se llevó a cabo la prueba de recuperación de la información de la cinta, utilizando el conector de salida para audífonos y el circuito demodulador, observando en el osciloscopio las señales de salida, nuevamente se verificaron sus frecuencias y amplitudes. Obteniéndose la recuperación íntegra de la señal de audio original.

Finalmente se repitió la prueba, pero ahora con el programa de control de la interfaz, un osciloscopio, el separador de sincronía vertical, el circuito modulador FSK, el circuito demodulador FSK, el control remoto adaptado y una cámara. La línea de transmisión del puerto serie de la PC se conectó al circuito modulador, y éste al conector de entrada para micrófono de las cámaras. La línea de recepción del puerto serie enlaza al circuito demodulador, y éste al conector de salida para audífonos de las cámaras. Se conectó el control remoto a la interfaz y ésta a su vez a los conectores de entrada para control remoto de las cámaras.

Para verificar la grabación de información en cada cuadro de imagen, el programa de control inicia la grabación y envía información única a cada cuadro. Se observan en el osciloscopio las señales que transmite el puerto serie, la salida del circuito modulador, y el pulso de sincronía vertical de una cámara, comprobándose que se transmiten a cada cuadro doce bytes de información según el protocolo establecido. Cada cuadro se identifica con dos pulsos de sincronía vertical de la cámara.

Para probar la lectura de información de cada imagen, el programa de control regresa las cintas, e inicia la reproducción de la cámara y despliega en pantalla la información leída de cada cuadro. Se observan en el osciloscopio las señales que recibe el puerto serie, de entrada al circuito demodulador y el pulso de sincronía vertical de la cámara. Se comprueba que se leen doce bytes de información por cada imagen a través del puerto serie.

5.3 Pruebas al mecanismo de posicionamiento y ajuste de distancia focal.

Antes de conectar el circuito de control a los servomotores, se verificó que la duración de los pulsos generados por el circuito de control fuera la necesaria para que los servomotores se muevan ± 45 grados, ya que si se sobrepasa este ángulo, el rotor se detiene por un tope mecánico, provocando un sobrecalentamiento que puede dañar permanentemente al servomotor. Una vez realizado lo anterior, se conectaron los servomotores y el mecanismo para evaluar el desempeño, obteniéndose resultados adecuados. Igualmente se verificó que la etapa de potencia que incluye el circuito de control, no interfiere en momento alguno en la operación de los demás componentes del sistema (cámaras, sensores, computadora); es decir, que no se detectaron efectos indeseables por causa de interferencia electromagnética espuria.

5.4 Pruebas a la adquisición de datos de sensores.

Se evaluó el comportamiento del convertidor analógico-digital en la recepción y conversión de datos provenientes de los sensores, así como el programa que controla la interfaz. Un sensor generalmente detecta cambios de una variable física y proporciona un nivel de voltaje que debe ser interpretado, para lo cual se utiliza un generador de funciones que simula cambios del nivel de voltaje. Cada valor recibido del convertidor analógico-digital se escala dentro de un intervalo de

valores apropiados a la variable en cuestión, y el resultado se muestra en pantalla.

Durante una hora se convirtieron y registraron 15 voltajes de diferente magnitud (hasta 9V), con estos datos se calcularon quince medias aritméticas, obteniéndose un error máximo del 5%; esto debido al tipo de capacitor empleado en el muestreador-retén y a la velocidad con que la computadora accesa las muestras de los voltajes de entrada.

También, se determinó que el tiempo que el programa espera es de 35 microsegundos, para que el convertidor analógico-digital y la electrónica implementada funcionen con la computadora adecuadamente. Dada la velocidad de operación del convertidor, a cada imagen se le asocian directamente los datos proporcionados por los sensores, es decir, que cada etiqueta incluye los datos de los sensores para facilitar su comportamiento subsecuente.

5.5 Pruebas de manejo del receptor GPS.

En este caso no hubo necesidad desarrollar una interfaz especial ya que el receptor de GPS recibe y transmite información por el puerto serie de la computadora, por lo que sólo se comprobó el programa de control correspondiente. Primero, se transmiten los comandos para la inicialización del receptor, además de enviar los comandos de acceso a la información requerida cada segundo. Después de los minutos requeridos se recibió cada segundo el bloque de información solicitada del receptor, igualmente se verificó que la fuera información fuera interpretada correctamente.

Además, se establecieron los tiempos de acceso de la computadora, para que el receptor GPS esté en condiciones de recibir un nuevo comando.

5.6 Discusión de resultados.

Las imágenes de video son etiquetadas utilizando un circuito externo a las cámaras, lo cual permite conocer la posición y orientación de los centros de toma de cada cuadro para su posterior interpretación. Durante la reproducción, la sincronización de las imágenes se realiza con base en una de las cámaras, lo que puede ocasionar un error inicial de defasamiento máximo entre las imágenes de las cámaras de hasta dos cuadros. Considerando que una velocidad de vuelo común de la aeronave durante la adquisición de imágenes es de 140 Km/hr, se tiene que la distancia entre los centros de toma de dos imágenes consecutivas es de 1.29 metros en el peor caso, y dado que el traslape entre imágenes útiles es generalmente del 60%, el error de localización es despreciable.

Las cámaras de video se controlan por computadora con circuitos externos y un control remoto adaptado, por lo que la transmisión de funciones a todas las cámaras del sistema se realiza secuencialmente.

El receptor GPS proporciona información de la geoposición del centro de toma de las imágenes cada segundo. Como la velocidad de grabación de las cámaras es de 30 imágenes por segundo, se requiere interpolar dicha información. Esto se hace trazando la ruta de vuelo de la aeronave para que cada imagen cuente con datos de geoposición de su centro de toma.

No obstante, dichos centros de toma presentan un error de hasta ± 50 metros (error estimado, proporcionado por el receptor GPS). Utilizando la técnica conocida como GPS diferencial, el error de posición del centro de toma es reducido hasta ± 3 metros, para lo cual se requieren dos receptores GPS. Desgraciadamente, para la realización de esta tesis, solo se contó con un receptor GPS, por lo que dicha precisión no pudo ser demostrada con pruebas.

6. CONCLUSIONES Y RECOMENDACIONES.

El objetivo principal de esta tesis fue, desarrollar un sistema aeroportado de adquisición de imágenes georreferenciadas, para su utilización en percepción remota. Con base en los resultados obtenidos, los autores concluimos:

- 1. Se desarrollaron los circuitos electrónicos y la programación necesarios para llevar a cabo campañas de teledetección con video multispectral para realizar estudios de percepción remota.**
- 2. Se integró de forma automática la georreferencia y orientación de los centros de toma de cada imagen.**
- 3. Con la posibilidad de corregir geoméricamente las imágenes y con la rápida formación de mosaicos, es posible llevar a cabo estudios de percepción remota en tiempos cortos (v.g. días o semanas).**
- 4. Se hace posible y costeable la actualización de mapas cartográficos con las imágenes de video adquiridas durante las campañas de detección, aumentando la resolución de dichos mapas.**
- 5. La presentación de información sobre el estado de los diferentes parámetros del sistema, al personal coordinador de vuelo, a través de un monitor a bordo, facilita el desarrollo de la campaña, ahorrando pérdidas innecesarias de imágenes y tiempo de vuelo, cuando algún parámetro crucial del sistema falla durante el vuelo.**

6. El control automático por computadora de las variables importantes del sistema, hace eficiente el desarrollo y aplicación de esta técnica de teledetección.
7. El número de sensores capaces de ser añadidos al sistema abre una amplia gama de aplicaciones. Múltiples variables ambientales pueden ser medidas, desde temperatura y presión atmosférica hasta humedad, así como diversos componentes químicos presentes en la atmósfera.

Un sistema como el desarrollado para esta tesis es susceptible de mejorarse. Las modificaciones que se proponen son:

1. Se recomienda aumentar la velocidad de grabación de datos en la pista de audio de las cámaras a 9600 bauds, lo que permitiría almacenar más información en las mismas, y en consecuencia se prescindiría del almacenamiento de información en el disco de la microcomputadora. Esta mejora daría mayor autonomía a las campañas de teledetección.
2. Un altímetro relativo al terreno basado en radiofrecuencia (radioaltímetro Doppler), representaría una mejora al sistema, y repercutiría en una mayor precisión en la interpretación y corrección geométrica de las tomas realizadas.
3. Para reducir los problemas de vibración inherentes a las aeronaves se recomienda utilizar cámaras de video con sistema de estabilización de imagen integrado.
5. Puede ser conveniente desarrollar un sistema de estabilización en tiempo real del contenedor de cámaras para ir corrigiendo sobre el vuelo las

perturbaciones dinámicas de la aeronave, con lo que se podría ahorrar mucho tiempo en el procesamiento de las imágenes.

APÉNDICE A: CARACTERÍSTICAS DEL RECEPTOR GPS.

El receptor del GPS empleado es un producto de la Magellan System Corporation.
El modelo es GPS OEM 5000 cuyas características más importantes son:

1. Características de Operación.

Razón de actualización de datos:	1 segundo continuo
Tiempo para el primer posicionamiento	
Arranque caliente:	30 segundos
Arranque frío:	72 segundos 2D/3D
Arranque autónomo:	8-12 minutos.
Velocidad máxima:	1529 kph
Aceleración máxima:	2 g

2. Exactitud de Posición (con modo de disponibilidad selectiva (S/A) inhabilitada) (HDOP < 2, CNo > 47 db-Hz, 2D)

Posición horizontal:	25 m rms en 2D, 30 m rms en 3D
Posición vertical	50 m rms
Velocidad	0.077 m/s rms

3. Características del Ambiente de Trabajo.

Temperatura de operación	-40 °C a +85 °C
Humedad relativa de operación	hasta 95% no condensado a 38 °C
Altitud de operación	-300 a +12 100 m

4. Características Eléctricas.

Potencia:	235 mA a 12 VDC típico, 9-16 VDC 250 mA máximo (sin considerar la antena)
Interfaces:	Un puerto de comunicación serial estándar industrial, configurado según RS-232, RS-422, NMEA-compatible o TTL-compatible.

APÉNDICE B: CARACTERÍSTICAS DE LA PISTA DE AUDIO DE LAS CINTAS.

La etiqueta o número identificador de cada imagen, se graba en la pista de audio de las cintas de video, a través de la entrada para micrófono de la cámara. La máxima amplitud de señal que puede enviarse a la entrada del micrófono es de 50 mV (voltaje de saturación). La respuesta en frecuencia se muestra en la siguiente gráfica, donde se puede apreciar que el canal de audio actúa como un filtro pasa bajas con una frecuencia de corte alrededor de los 10 kHz.

Para conocer la respuesta en amplitud y frecuencia del canal de audio de la cámara y cinta de video, se llevó a cabo una evaluación por medio de un generador de funciones, una cámara y un osciloscopio de dos canales. Primero, se utilizó una señal de 1 kHz variando su amplitud hasta determinar el voltaje de saturación que fue de 50 mV. Posteriormente, se alimentó una señal de amplitud fija y menor que el voltaje de saturación, variando su frecuencia de 100 Hz hasta 20 kHz obteniéndose la gráfica de la figura 32.

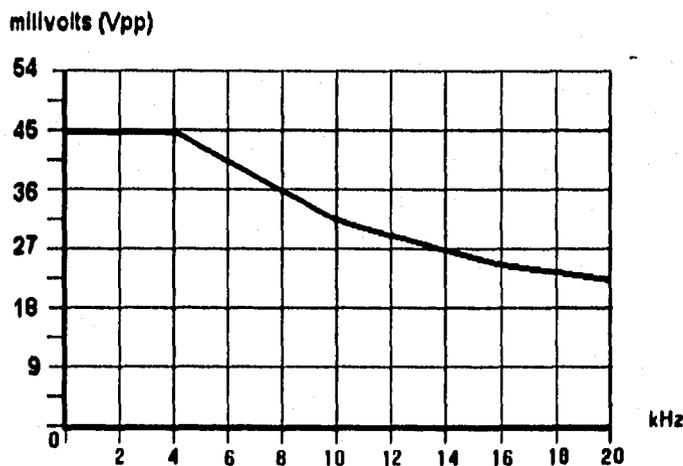


Figura 33. Gráfica de la respuesta en frecuencia del micrófono de la cámara.

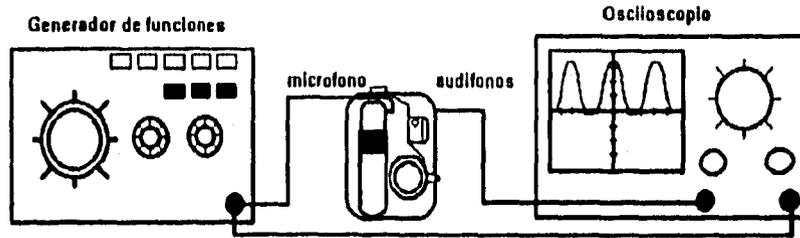


Figura 34. Experimento para conocer la operación del micrófono de la cámara.

La información para cada imagen se transmite por el puerto serie de la microcomputadora, esta señal es una secuencia de unos y ceros en formato RS-232 (-12V para un uno lógico y +12V para un cero lógico), con componentes de alta frecuencia que están fuera del ancho de banda que registra el canal de audio. Por esta razón se diseñó un circuito de codificación por corrimiento en frecuencia (FSK, Frequency Shift Keying), así la señal binaria es transformada en una senoide con una amplitud fija de 45 mV y dos frecuencias (6.1 kHz define un uno lógico y 8.5 kHz define un cero lógico). Las señales FSK presentan alta inmunidad al ruido, y dado que el sistema operará en un ambiente de alta interferencia electromagnética, éstas resultan las adecuadas; además de operar en el intervalo adecuado al canal de audio de las cámaras.

La transmisión se realiza a 4800 bits por segundo, posibilitando grabar hasta 13.2 bytes de información en cada imagen. Se estableció el siguiente protocolo: un byte de inicio, ocho bytes de información y 4.2 bytes no se transmiten y proporcionan tiempo en la recuperación información.

APÉNDICE C: EL PLL (Phase Locked Loop)

Una malla de fase encadenada (PLL, Phase Locked Loop) es un circuito electrónico que se compone de un detector de fase, un filtro pasobajas y un oscilador controlado por voltaje (VCO), conectados como se muestra a continuación:

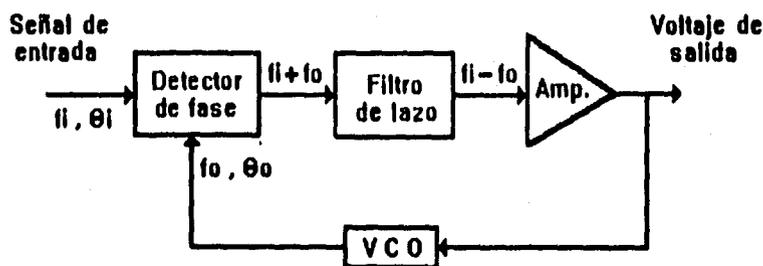


Figura 35. Diagrama de bloques de un ciclo de amarre de fase (PLL).

La operación básica de un circuito PLL puede explicarse empleando como referencia la figura anterior. La señal de entrada y la proveniente del VCO se comparan mediante el detector de fase, su voltaje de salida es proporcional a la diferencia de fase entre las dos señales; este voltaje pasa por un filtro pasobajas y después por el amplificador, el cual se emplea para modular la frecuencia del VCO. La configuración de lazo cerrado del circuito es para mantener la frecuencia del VCO amarrada a la frecuencia de la señal de entrada.

Debido al intervalo limitado de operación del VCO y a la conexión de retroalimentación del circuito PLL, existen dos bandas de frecuencias importantes que se especifican para un PLL. La *banda de captura* de un PLL se define como el intervalo en que la frecuencia libre de oscilación del VCO se amarra a la señal de entrada. Una vez que el PLL ha "capturado" la señal de entrada, se puede

mantener encadenado a dicha señal sobre un intervalo de frecuencias denominado *banda de encadenamiento*.

El PLL puede utilizarse en una amplia variedad de aplicaciones, entre las que se incluyen:

- Demodulación y síntesis de frecuencia.
- Decodificación FSK.
- Redes de demodulación FM.
- Detectores de AM.
- Filtros de rastreo.
- Receptores y transmisores de telemetría.

Las ecuaciones para encontrar los valores de resistencias y capacitores, para el circuito integrado utilizado son las siguientes:

La frecuencia libre de oscilación (f_0) se eligió a la mitad de las frecuencias de marca y espacio, el valor del capacitor C_0 también se conoce.

$$R_0 = \frac{0.3}{C_0 f_0} \quad , \quad R_0 = R_t + P_t$$

El filtro de lazo que se empleó fue el de adelanto-atraso, en este caso los parámetros conocidos son la frecuencia natural del lazo (f_n), en este caso de 2400 Hz y el factor de amortiguamiento (δ) que para una respuesta transitoria, se recomienda de 0.5 a 1, el valor elegido fue de 0.7, ya que es valor óptimo.

En esta aplicación es utilizado un circuito para comparar el nivel de DC del filtro paso bajas con un voltaje de referencia y así poder generar los dos niveles de voltaje de salida del PLL correspondientes a las dos frecuencias de la señal FSK.

$$K_o K_d = \frac{33.6 f_o}{V_c}$$

$$C_f = \frac{K_o K_d}{R_1 (2 \pi f_n)} \cdot \frac{\delta}{R_1 \pi f_n}$$

$$R_f = \frac{\delta}{\pi f_n C_f}$$

APÉNDICE D: LENGUAJES DE PROGRAMACIÓN.

Antes de desarrollar el programa de control, fué necesario seleccionar el lenguaje que cumpliera con ciertas especificaciones, a saber:

- a) permitir manipular de manera directa direcciones de memoria,
- b) posibilidad para evaluar datos a nivel de bits,
- c) facilidad para adecuar el funcionamiento de diferentes componentes del equipo de cómputo (tales como puertos de comunicación, posibilidad de alterar y definir rutinas de servicios de interrupción, etc.).

De acuerdo al tipo de herramientas que proporcionan se identifican tres grupos de lenguajes: de alto nivel, de nivel medio y de bajo nivel.

Entre los lenguajes de alto nivel se encuentran ADA, MODULA-2, PASCAL, COBOL, FORTRAN y BASIC. Como todos sabemos, un lenguaje de alto nivel proporciona herramientas preconstruidas que facilitan la programación de ciertas aplicaciones; sin embargo, dichas herramientas son específicas y monolíticas, lo que redundo en una inflexibilidad en la programación y en consecuencia una limitación en el tipo de aplicaciones que se pueden desarrollar, sobre todo cuando se trabaja con interfaces como en este caso.

Los lenguajes de bajo nivel los constituyen los ENSAMBLADORES de los microprocesadores. Un lenguaje de bajo nivel no proporciona ninguna herramienta preconstruida, se programa bajo mnemónicos y se trabaja directamente con los registros del microprocesador y las direcciones de memoria. Esto permite una flexibilidad total, sin embargo, se requiere desarrollar todo, incluyendo las rutinas de control de los componentes inherentes a un equipo de cómputo.

Entre los lenguajes de nivel intermedio tenemos a C, FORTH y MACRO-ASSEMBLER. Un lenguaje de nivel medio posee herramientas preconstruidas que facilitan la programación a la vez que permite la manipulación directa de bits, bytes, direcciones de memoria y registros del microprocesador. Lo anterior permite una alta flexibilidad en la programación de aplicaciones y la facilidad de llevar a cabo el control de interfaces electrónicas como si fuera un ensamblador.

Dado lo anterior, se seleccionó el lenguaje "C" para desarrollar el programa de control de las interfaces electrónicas desarrolladas. C es un lenguaje de nivel medio porque combina elementos de un lenguaje de alto nivel con la funcionalidad de un lenguaje ensamblador. En especial C permite la manipulación directa de bits, bytes, palabras y apuntadores.

APÉNDICE E: CÓDIGO DEL PROGRAMA DE CONTROL

En las páginas siguientes se lista el código del programa de control del sistema. Cabe aclarar que fueron suprimidas algunas líneas importantes del código fuente del programa con el fin de proteger la autoría del mismo. Sin embargo, se proporciona el programa ejecutable.

```

.....
TESIS: SISTEMA AEROPORTADO PARA OBTENER IMAGENES DE VIDEO
PROGRAMA DE CONTROL DEL SISTEMA DE ADQUISICION DE IMAGENES

DESARROLLARON: RUBEN DOMINGUEZ TREJO
MARCO ANTONIO HERNANDEZ GUTIERREZ
JAIME CONTRERAS CASTRO

DIRECTORA: M. en C. MARGARITA NAVARRETE MONTESIMOS

LABORATORIO DE INGENIERIA AEROSPAZIAL. INSTITUTO DE INGENIERIA. UNAM.
ENERO 1996
.....

```

```

EL PRESENTE PROGRAMA CONTROLA, COMUNICA Y SUPERVISA EL BUEN FUNCIONAMIENTO
DE TODOS LOS COMPONENTES DEL SISTEMA. EL PROGRAMA MANTIENE UNA COMUNICACION
CONSTANTE CON EL USUARIO PARA INDICARLE EL ESTADO DE OPERACION DEL SISTEMA.
EL PROGRAMA AUTOMATIZA COMPLETAMENTE TODOS LOS PROCESOS DEL SISTEMA
LIMITANDO LA INTERVENCION DIRECTA DEL USUARIO AL MINIMO INDISPENSABLE.

```

```

NOTA: EL PROGRAMA SE ENCUENTRA INCOMPLETO COMO MEDIDA DE PROTECCION.
.....

```

```

#include <dos.h>
#include <conio.h>
#include <stdlib.h>
#include <bios.h>
#include <stdio.h>
#include <math.h>
#include <string.h>
#include <io.h>
#include <ctype.h>

```

----- DEFINICION DE VALORES

```

#define Convert 0x300 // DIRECCION DEL CONVERTIDOR ANALOGICO DIGITAL
#define Control 0x302 // DIRECCION DEL CONTROL REMOTO DE LAS CAMARAS
#define Ncamara 0x303 // DIRECCION PARA ELEGIR UNA DE LAS CAMARAS
#define Noservo 0x305 // DIRECCION PARA ELEGIR UNO DE LOS SERVO MOTORES
#define Mvservo 0x307 // DIRECCION DE MOVIMIENTO PARA SERVO MOTORES
#define Censable 0x308 // DIRECCION PARA SENSAR ESTADO RECORD DE CAMARAS
#define Nsensor 0x30C // DIRECCION PARA ELEGIR UNO DE LOS SENSORES

#define ESC 27 // ESCAPE

#define OFF 0 // SIN OPERACION MODOS DE OPERACION DEL SISTEMA
#define VTR 1 // OPERACION EN TIERRA
#define CAM 2 // OPERACION EN VUELO
#define REC 3 // OPERACION DE GRABACION DE IMAGENES
#define GPS 4 // OPERACION DEL GPS

#define Off 0x00 // CAMARA APAGADA FUNCIONES DE CAMARA
#define Rec 0x01 // GANBA CINTAS
#define Stby 0x11 // CINTAS EN ESPERA
#define Fast 0x08 // ADELANTA CINTAS
#define Stop 0x10 // DETIENE CINTAS
#define Rew 0x20 // RETRASA CINTAS
#define Play 0x40 // REPRODUCE CINTAS
#define Pausa 0x80 // CINTAS EN PAUSA

```

```

#define ServoX 0x0B // SERVO MOTOR PARA EJE X
#define ServoY 0x0F // SERVO MOTOR PARA EJE Y
#define SrZoom 0x0D // SERVO MOTOR PARA ZOOM

#define ZoomMax 0xFF // ZOOM MAXIMO DE CAMARA MOVIL
#define ZoomMin 0x00 // ZOOM MINIMO DE CAMARA MOVIL

#define GPZDA 0x01 // A00: DIA Y HORA MENSAJES ENVIADOS POR EL GPS
#define GPGGA 0x02 // B00: LATITUD Y LONGITUD
#define PMSLB 0x03 // B02: ALTITUD EXTENDIDA
#define GPVTG 0x04 // E00: CURSO Y VELOCIDAD
#define PMSLF 0x05 // F02: SATELITES UTILIZADOS
#define PMSLG 0x06 // G00: ERROR ESTIMADO
#define PMSLH 0x07 // H00: ESTADO DEL GPS

```

----- TERMINA DEFINICION DE VALORES

----- DEFINICION FUNCIONES DEL SISTEMA

```

/* FUNCIONES QUE CONTROLAN LA INTERFAZ AL USUARIO DE DESPLIEGUE DE
INFORMACION EN PANTALLA */
void scroll(int x1,int y1,int x2,int y2,int fondo);
void veaxy(int x,int y);
void readch(unsigned* letra,unsigned* atrib);
void writch(int letra,int atrib);
void marco(int x1,int y1,int x2,int y2,int frente,int fondo);
void print(int x, int y, char texto[40], int frente, int fondo);
void startdisplay(void);
void workdisplay(char *tipo);
void comenta(char *texto, int avanza);
void blinking(unsigned blink, int col, int ren, int size, unsigned* opcion);

/* FUNCIONES DE CONTROL GENERALES DEL SISTEMA GENERALES */
void loadato(char cadena[12], int size, char tipo[5]);
void abrefile(char file[12], FILE **pfile);
void closefile(char file[12], FILE **pfile);
void puertoserie(int port, int baud, int word, int ints, int modo);
void startvar(unsigned char modo);
void displaydatos(unsigned char tipo);
void ctrlcamara(unsigned char orden);

/* FUNCIONES DE CONTROL EXCLUSIVAS DEL MODO DE TIERRA (VTR) */
void vtr(void);
void asciifile(char file[12], FILE **pfile);
void finddato(char tipo[7], long valor, FILE **pfile);
void gotonoimg(long imagen);
void digitaliza(void);

/* FUNCIONES MODO DE CONTROL EXCLUSIVAS DE VUELO (CAM) */
void cam(void);
void tmsg(char *mensaje);
int revisachksun(char mensaje[100]);
void generachksun(char mensaje[100], unsigned char chksun[3]);

void camrec(void);
void almaceca(FILE **pfile);
void sensor(unsigned char sen, unsigned* word);
void procesamsgps(char mensaje[100]);
void moveservo(unsigned servo, unsigned grados);
void ctrlservo(unsigned col, unsigned ren, unsigned zoom);
void enablecam(void);

void campps(void);

```

```

void camgps(void);
void startgps(void);
void cleargps(void);
void comandogps(void);

//----- TERMINA DEFINICION DE FUNCIONES DEL SISTEMA
//----- DEFINICION DE VARIABLES

unsigned CAMARAS=1, SENSORES=1; // CAMARAS Y SENSORES EN EL SISTEMA

struct { // REGISTRO DE DATOS DE SENSORES Y GPS
    long noimagen; unsigned alabeo, cabeceo, brujula;
    float latitud, longitud; long altitud, hora; int error; } reg;

struct { // ESTADO DE OPERACION DE CADA CAMARA
    unsigned char code; char *name; } fcam[4];

struct { // VARIABLES PARA EL CONTROL DEL GPS
    unsigned numsat[5], status;
    char edosat[5], *fecha, *hora; float curso, speed; } vgps;

union REGS scr; // REGISTRO PARA MANEJAR EL VIDEO
union { char ch[2]; int i; } key; // PARA EL MANEJO DE TECLA PRESIONADA

unsigned ren, col; // RENGLON Y COLUMNA, VENTANA DE COMENTARIOS
unsigned renfun, colfun; // RENGLON Y COLUMNA, VENTANA DE FUNCIONES
unsigned indice; // VARIABLE DE CONTEO
unsigned letra, atrib; // CARACTER Y ATRIBUTOS EN EL CURSOR
unsigned opcio; // OPCION SELECCIONADA DE MENU DE FUNCIONES

size_t wordreg=sizeof(reg); // LONGITUD DEL REGISTRO DE DATOS

char *file; FILE *pfile; // ARCHIVO BINARIO DE DATOS

int MODO-OFF; // MODO DE OPERACION

long imgcam[4]; // NO. DE IMAGEN EN CADA CAMARA
float vsensor[16]; // VALORES DE LOS SENSORES EN EL SISTEMA

long nocuadro; // NO. DE IMAGEN DE LAS CAMARAS
unsigned byte1, byte2, byte3; // BYTES DEL NO. DE IMAGEN
unsigned nobyteimg; // NO. DE BYTE DEL NO. DE IMAGEN

long despierta; // BANDERA PARA DESPERTAR CAMARAS

unsigned pulso; // PULSO VERTICAL PARA SINCRONIZAR IMAGENES

unsigned newmsggps; // BANDERA DE NUEVO MENSAJE DEL GPS
unsigned newreggps; // BANDERA DE NUEVO REGISTRO DEL GPS
unsigned nomsggps; // NO. DE MENSAJE RECIBIDO DEL GPS
int chrgps; // NO. DE CARACTER DEL MENSAJE DEL GPS
char msggps1[100], msggps2[100]; // MENSAJES RECIBIDOS DEL GPS

unsigned comuta; // BANDERA DE COMUTACION DE CANAL
unsigned camara; // COMUTACION DE CAMARA

//----- TERMINA DEFINICION DE VARIABLES

/* -----
FUNCIONES QUE CONTROLAN LA INTERFAZ AL USUARIO DE DESPLIEGUE DE
INFORMACION EN PANTALLA
-----*/

```

```

/* GENERA UNA VENTANA DE TEXTO EN PANTALLA CON UN COLOR DE FONDO DEFINIDO
UTILIZANDO LOS SERVICIOS DE INTERRUPCION DEL BIOS DE LA COMPUTADORA
(X1, Y1), COLUMNA Y RENGLON DE LA ESQUINA SUPERIOR IZQUIERDA
(X2, Y2), COLUMNA Y RENGLON DE LA ESQUINA INFERIOR DERECHA
FONDO, DEFINE EL COLOR DE FONDO DE LA VENTANA */

void scroll(int x1,int y1,int x2,int y2,int fondo)
{
    x1--; y1--; x2--; y2--; // AJUSTA COORDENADAS A PANTALLA
    scr.h.ah=6; scr.h.al=0; // CODIGO DE LA ORDEN DE LA INTERRUPCION
    scr.h.ch=y1; scr.h.cl=x1; // COORDENADAS ESQUINA SUPERIOR IZQUIERDA
    scr.h.dh=y2; scr.h.dl=x2; // COORDENADAS ESQUINA INFERIOR DERECHA
    scr.h.bh=fondo<<4; // COLOR DEL FONDO DE LA VENTANA
    int86(0x10,6scr,6scr); // INTERRUPCION DE VIDEO
    return;
} // TERMINA SCROLL

/* POSICIONA EL CURSOR EN LA POSICION INDICADA EN PANTALLA
(X, Y) COLUMNA Y RENGLON DE LA POSICION DEL CURSOR */
void veaxy(int x,int y)
{
    scr.h.ah=2; scr.h.bh=0; // CODIGO DE LA ORDEN DE LA INTERRUPCION
    scr.h.dh=y; scr.h.dl=x; // COORDENADAS DE LA POSICION DEL CURSOR
    int86(0x10,6scr,6scr); // INTERRUPCION DE VIDEO
    return;
} // TERMINA VEAXY

/* LEE EL CARACTER Y SUS ATRIBUTOS EN LA POSICION DEL CURSOR EN PANTALLA
LETRA, VARIABLE EN QUE SE DEJA EL CARACTER LEIDO
ATRIB, VARIABLE EN QUE SE DEJAN LOS ATRIBUTOS DEL CARACTER LEIDO */
void readch(unsigned* letra,unsigned* atrib)
{
    scr.h.ah=8; scr.h.bh=0; // CODIGO DE LA ORDEN DE LA INTERRUPCION
    int86(0x10,4scr,6scr); // INTERRUPCION DE VIDEO
    *atrib=scr.h.ah; // RECOGE LOS ATRIBUTOS DEL CARACTER
    *letra=scr.h.al; // RECOGE EL CARACTER
    return;
} // TERMINA WRITE

/* ESCRIBE UN CARACTER CON LOS ATRIBUTOS INDICADOS EN LA POSICION DEL
CURSOR EN PANTALLA
LETRA, CARACTER A ESCRIBIR EN PANTALLA
ATRIB, ATRIBUTOS DEL CARACTER A ESCRIBIR EN PANTALLA */
void writech(int letra,int atrib)
{
    scr.h.ah=9; // CODIGO DE LA ORDEN DE LA INTERRUPCION
    scr.h.al=letra; // CARACTER A ESCRIBIR EN PANTALLA
    scr.h.bh=0; scr.h.bl=atrib; // ATRIBUTOS DEL CARACTER A ESCRIBIR
    scr.h.ch=0; scr.h.cl=1; // NUMERO DE CARACTERES A ESCRIBIR
    int86(0x10,4scr,4scr); // INTERRUPCION DE VIDEO
    return;
} // TERMINA WRITE

/* DIBUJA EN PANTALLA UNA CAJA EN LA POSICION Y CON LOS ATRIBUTOS INDICADOS
(X1, Y1), COLUMNA Y RENGLON DE LA ESQUINA SUPERIOR IZQUIERDA
(X2, Y2), COLUMNA Y RENGLON DE LA ESQUINA INFERIOR DERECHA
FRENTE, COLOR DEL CONTORNO DE LA CAJA
FONDO, COLOR DEL FONDO DEL CONTORNO DE LA CAJA */
void marco(int x1,int y1,int x2,int y2,int frente,int fondo)
{
    int indice; // VARIABLE PARA CICLO DE CONTEO
    int e1=218, e2=191, e3=192, e4=217; // CARACTERES DE LAS ESQUINAS

```

```

int lh=196, lv=179: // CARACTERES DE LOS LADOS DE LA CAJA
unsigned atrib: // ATRIBUTOS DEL CONTORNO DE LA CAJA

atrib=(fondo<<4)|frente: // CONTRUYE PALABRA DE ATRIBUTOS
xl--: yl--: x2--: y2--: // AJUSTA COORDENADAS A LA PANTALLA

// DIBUJA ESQUINAS DE LA CAJA
veaxy(x1,y1): writtech(e1,atrib): veaxy(x2,y1): writtech(e2,atrib):
veaxy(x1,y2): writtech(e3,atrib): veaxy(x2,y2): writtech(e4,atrib):

for(indice=x1+1: indice<=x2-1: indice++) { // DIBUJA LADOS HORIZONTALES
veaxy(indice,y1): writtech(lh,atrib):
veaxy(indice,y2): writtech(lv,atrib): }

for(indice=y1+1: indice<=y2-1: indice++) { // DIBUJA LADOS VERTICALES
veaxy(x1,indice): writtech(lv,atrib):
veaxy(x2,indice): writtech(lh,atrib): }

return:
} // TERMINA MARCO

/* ESCRIBE UNA MENSAJE EN PANTALLA EN LA POSICION Y CON LOS ATRIBUTOS
INDICADOS
(X, Y), COLUMNA Y RENGLON DE INICIO DEL MENSAJE A ESCRIBIR
TEXTO, MENSAJE A ESCRIBIR
FRENTE, COLOR DE LOS CARACTERES DEL MENSAJE
FONDO, COLOR DEL FONDO DE LOS CARACTERES DEL MENSAJE
void print(int x, int y, char texto[40], int frente, int fondo)
{
int indice: unsigned atrib:

x--: y--: // AJUSTA COORDENADAS
atrib=(fondo<<4)|frente:
for(indice=0: texto[indice]!=0x00: indice++) {
veaxy(x+indice,y): writtech(texto[indice],atrib):
}
return:
} // TERMINA PRINT

// DIBUJA LA PANTALLA DE INICIO DEL SISTEMA */
void startdisplay(void)
{
// DIBUJA EL TAPIZ DE FONDO
for(ren=2: ren<=24: ren+=3) {
for(col=1: col<=79: col+=6) print(col,ren+0,"",WHITE,BLUE):
for(col=1: col<=79: col+=6) print(col,ren+1,"",WHITE,BLUE):
for(col=1: col<=79: col+=6) print(col,ren+2,"",WHITE,BLUE):
}

// DIBUJA TITULO SUPERIOR
scroll(1,1,80,1,LIGHTGRAY):
print(2,1,"Laboratorio de Ingenieria Aeroespacial.",BLACK,LIGHTGRAY):
print(46,1,"Instituto de Ingenieria, U.N.A.M.",BLACK,LIGHTGRAY):

// DIBUJA MENU DE INICIO
scroll(57,4,76,8,LIGHTGRAY): // DEFINE VENTANA
marco(57,4,76,8,BLACK,LIGHTGRAY): // DIBUJA CAJA
print(59,4,"Operación",BLACK,LIGHTGRAY):
print(59,5,"1. Vuelo (CAM)",BLACK,LIGHTGRAY):
print(59,6,"2. Tierra (VTR)",BLACK,LIGHTGRAY):
print(59,7,"0. Salir.",BLACK,LIGHTGRAY):

marco(55,10,76,12,BLACK,LIGHTGRAY): // DIBUJA CAJA DE AYUDA EN LINEA

```

```

// DIBUJA CREDITOS
marco(5,14,42,18,BLACK,LIGHTGRAY): // DIBUJA CAJA
print(7,14,"Desarrollaron",BLACK,LIGHTGRAY):
print(6,15,"Rubén Domínguez Trejo.",BLACK,LIGHTGRAY):
print(6,16,"Marco Antonio Hernández Gutiérrez.",BLACK,LIGHTGRAY):
print(6,17,"Jaime Contreras Castro.",BLACK,LIGHTGRAY):

marco(5,20,48,22,BLACK,LIGHTGRAY): // DIBUJA CAJA
print(7,20,"Dirigido",BLACK,LIGHTGRAY):
print(6,21,"W. en C. Margarita Navarrete Montesinos.",BLACK,LIGHTGRAY):

// DIBUJA TITULO INFERIOR
scroll(1,25,80,25,LIGHTGRAY):
print(2,25,"Sistema de Adquisición de Imágenes.",BLACK,LIGHTGRAY):
print(71,25,"Prototipo.",BLACK,LIGHTGRAY): veaxy(74,10):

return:
} // TERMINA STARTDISPLAY

/* DIBUJA PANTALLAS DE LOS DIFERENTES MODOS DE OPERACION DEL SISTEMA */
void workdisplay(char *tipo)
{
// DIBUJA EL TAPIZ DE FONDO
for(ren=2: ren<=24: ren+=3) {
for(col=1: col<=79: col+=6) print(col,ren+0,"",WHITE,BLUE):
for(col=1: col<=79: col+=6) print(col,ren+1,"",WHITE,BLUE):
for(col=1: col<=79: col+=6) print(col,ren+2,"",WHITE,BLUE):
}

// DIBUJA TITULO SUPERIOR
scroll(1,1,80,1,LIGHTGRAY):
print(2,1,"Sistema de Adquisición de Imágenes.",BLACK,LIGHTGRAY):

if(strstr(tipo,"vtr")!=NULL) { // DIBUJA VENTANAS: OPERACION EN TIERRA
print(60,1,"Operación en Tierra.",BLACK,LIGHTGRAY):
marco(1,2,40,10,WHITE,BLUE): print(3,2,"Controles",YELLOW,BLUE):
marco(41,2,60,4,WHITE,BLUE): print(43,2,"Fecha Archivo",YELLOW,BLUE):
}
else { // DIBUJA VENTANAS: OPERACION EN VUELO
print(61,1,"Operación en Vuelo.",BLACK,LIGHTGRAY):
marco(1,2,25,5,WHITE,BLUE):
print(3,2,"Sat Sat Sat Sat Sat",YELLOW,BLUE):
marco(26,2,40,4,WHITE,BLUE): print(28,2,"Status GPS",YELLOW,BLUE):
marco(41,2,60,4,WHITE,BLUE): print(43,2,"Fecha GRT",YELLOW,BLUE):

marco(17,5,40,13,WHITE,BLUE):
print(19,5,"Imagen Camara Móvil",YELLOW,BLUE):
marco(1,6,25,12,WHITE,BLUE): print(3,6,"Controles",YELLOW,BLUE):

marco(41,11,53,13,WHITE,BLUE): print(43,11,"Mov.X",YELLOW,BLUE):
marco(54,11,66,13,WHITE,BLUE): print(56,11,"Mov.Y",YELLOW,BLUE):
marco(67,11,80,13,WHITE,BLUE): print(69,11,"Mov.Zoom",YELLOW,BLUE):
}

// DIBUJA VENTANAS EN PANTALLA COMUNES A MODOS DE OPERACION EN VUELO
Y EN TIERRA
marco(61,2,80,4,WHITE,BLUE): print(63,2,"Hora GMT",YELLOW,BLUE):
marco(41,5,60,7,WHITE,BLUE): print(43,5,"Latitud",YELLOW,BLUE):
marco(61,5,80,7,WHITE,BLUE): print(63,5,"Longitud",YELLOW,BLUE):

marco(41,8,60,10,WHITE,BLUE): print(43,8,"Altitud",YELLOW,BLUE):
marco(61,8,80,10,WHITE,BLUE): print(63,8,"Error GPS",YELLOW,BLUE):

```

```

marco( 1,14,40,19,WHITE,BLUE): print( 3,14,"Comentarios",YELLOW,BLUE);
marco(41,14,80,16,WHITE,BLUE):
print(43,14,"EdoCam1 EdoCam2 EdoCam3 EdoCam4",YELLOW,BLUE);
marco(41,17,80,19,WHITE,BLUE):
print(43,17,"ImgCam1 ImgCam2 ImgCam3 ImgCam4",YELLOW,BLUE);

marco( 1,20,40,22,WHITE,BLUE):
print( 3,20,"[Sensor] [Sensor] [Sensor] [Sensor]",YELLOW,BLUE);
marco( 1,22,40,24,WHITE,BLUE):
print( 3,22,"[Sensor] [Sensor] [Sensor] [Sensor]",YELLOW,BLUE);
marco(41,20,80,22,WHITE,BLUE):
print(43,20,"[Sensor] [Sensor] [Sensor] [Sensor]",YELLOW,BLUE);
marco(41,22,80,24,WHITE,BLUE):
print(43,22,"[Sensor] [Sensor] [Sensor] [Sensor]",YELLOW,BLUE);

// DIBUJA TITULO INFERIOR
scroll( 1,25,80,25,LIGHTGRAY):
print( 2,25,"Laboratorio de Ingenieria Aeroespacial.", BLACK,LIGHTGRAY);
print(47,25,"Instituto de Ingenieria, U.M.A.M.", BLACK,LIGHTGRAY);

return;
} // TERMINA WORKDISPLAY

/* MANEJA EL DESPLIEGUE DE INFORMACION EN LA VENTANA DE COMENTARIOS DURANTE
LA OPERACION DEL SISTEMA
TEXTO, INFORMACION A DESPLEGAR
AVANZA, BANDERA DE SEÑALIZACION PARA SALTAR O NO RENGLON */
void comenta(char *texto, int avanza)
{
window( 2,15,39,18); // SALTA A VENTANA DE COMENTARIOS
gotoxy(col,ren); // ULTIMA POSICION DEL CURSOR

// ESCRIBE EL MENSAJE CARCTER POR CARACTER
for(:"*texto;+texto) {
if(col>38) | ren++; col=1; // VIGILA FIN DE RENGLON
if(ren> 4) | // VIGILA FIN DE VENTANA
gotoxy(1,1); delline(); // BORRA PRIMER RENGLON DE VENTANA
scroll( 2,18,39,18,BLUE); // LIMPIA ULTIMO RENGLON DE VENTANA
col=1; ren=4; // DEFINE ULTIMO RENGLON

gotoxy(col++,ren); cprintf("%c",*texto); // DESPLIEGA CARACTER
}

if(avanza) | ren++; col=1; // AVANZA AL SIGUIENTE RENGLON

if(col>38) | ren++; col=1; // VIGILA FIN DE RENGLON
if(ren> 4) | // VIGILA FIN DE VENTANA
gotoxy(1,1); delline(); // BORRA PRIMER RENGLON DE VENTANA
scroll( 2,18,39,18,BLUE); // LIMPIA ULTIMO RENGLON DE VENTANA
col=1; ren=4; // DEFINE ULTIMO RENGLON

window( 1, 1,80,25); // VUELVE A PANTALLA COMPLETA
return;
} // TERMINA COMENTA

/* MANEJA EL DESTELLO DE LAS OPCIONES DE LOS MENUS DE FUNCIONES
BLINK, BANDERA PARA HACER O DESHACER EL DESTELLO
(COL, REN), COLUMNA Y RENGLON DE INICIO DE LA OPCION DE MENU
SIZE, TAMAÑO DE LA OPCION DE MENU. NUMERO DE CARACTERES
OPCION, ENTREGA LA OPCION DE MENU LEIDA PARA SU NOMBRE */
void blinking(unsigned blink, int col, int ren, int size, unsigned* opcion)
{
unsigned indice, letra, atrib;

```

```

// GENERA O ELIMINA EL DESTELLO DE LA OPCION CARACTER POR CARACTER
for(indice=col; indice<=(col+size); indice++) {
gotoxy(indice,ren); readch(&letra,&atrib); // LEE ATRIBUTOS Y CARACTER
if(blink==1) writch(letra, BLINK&atrib); // GENERA DESTELLO
else writch(letra,-BLINK&atrib); // ELIMINA DESTELLO
if(indice==col) *opcion=letra; // GUARDA OPCION LEIDA
}
return;
} // TERMINA BLINKING

/* =====
TERMINAN FUNCIONES QUE CONTROLAN LA INTERFACE AL USUARIO DE DESPLIEGUE DE
INFORMACION EN PANTALLA
===== */

/* =====
FUNCIONES GENERALES DEL SISTEMA
===== */

/* CONFIGURA EL PUERTO SERIE DE LA COMPUTADORA
PORT, PUERTO SERIE A CONFIGURAR
BAUD, VELOCIDAD DE OPERACION DEL PUERTO SERIE
WORD, FORMATO DE LAS PALABRAS DEL PUERTO SERIE
INTS, PERMITE O NO INTERRUPCIONES
MODO, MODO DE OPERACION DEL PUERTO SERIE */
void puertoserie(int port, int baud, int word, int ints, int modo)
{
unsigned char byte2,byte4,byte5,byte6; // DATOS PARA PUERTO
unsigned dir1,dir2,dir3,dir4,dir5,dir6; // DIRECCIONES PARA PUERTO

if(port==1) | // DIRECCIONES DEL PUERTO SERIE 1
dir1=0x3fb; dir2=0x3f8; dir3=0x3f9;
dir4=0x3fb; dir5=0x3fc; dir6=0x3f9; |
else | // DIRECCIONES DEL PUERTO SERIE 2
dir1=0x2fb; dir2=0x2f8; dir3=0x2f9;
dir4=0x2fb; dir5=0x2fc; dir6=0x2f9; |

if(baud==2400) byte2=0x30; // VELOCIDAD: 2400 BAUDS
else byte2=0x0c; // 9600 BAUDS

if(word==1) byte4=0x03; // SIN PARIDAD, 1 STOPBIT, 8 DATABITS
else byte4=0x1f; // PARIDAD PAR, 2 STOPBITS, 8 DATABITS

if(ints==1) byte5=0x00; // BLOQUEA INTERRUPCIONES
else byte5=0x08; // PERMITE INTERRUPCIONES

if(modo==1) byte6=0x00; // BLOQUEA INTERRUPT ENABLE REGISTER
else { if(modo==2) byte6=0x01; // ACTIVA RECEIVER DATA REGISTER
else byte6=0x02; // ACTIVA TRANSMITTER HOLDING REGISTER
}

delay(1); outportb(dir1, 0x00); // ACCESA REGISTRO DE BAUDAJE
delay(1); outportb(dir2,byte2); // DEFINE BAUDAJE
delay(1); outportb(dir3, 0x00); //
delay(1); outportb(dir4,byte4); // PARIDAD, STOPBITS, DATABITS
delay(1); outportb(dir5,byte5); // INTERRUPCIONES
delay(1); outportb(dir6,byte6); // ACTIVA REGISTROS DE INTERRUPCION
return;
} // TERMINA PUERTOSERIE

/* MANEJA LA LECTURA DE DATOS POR EL TECLADO SOLICITADOS AL USUARIO EN LA
VENTANA DE COMENTARIOS
CADENA, DATO A LEER

```

```

SIZE, NUMERO DE CARACTERES DEL DATO A LEER
TIPO, TIPO DE DATO A LEER
void leedato(char cadena[12], int size, char tipo[5])
{
    int indice=0;          // PARA CICLO DE CONTEO

    window( 2,15,39,18); // SALTA A VENTANA DE COMENTARIOS
    gotoxy(col,ren);      // SALTA A ULTIMA POSICION DEL CURSOR

    do {
        if(col>38) { ren++; col=1; } // VIGILA FIN DE RENGLON
        if(ren>4 ) {                // VIGILA FIN DE VENTANA
            gotoxy(1,1); delline(); // BORRA PRIMER RENGLON DE LA VENTANA
            scroll( 2,18,39,18,BLUE); // LIMPIA ULTIMO RENGLON DE LA VENTANA
            col=1; ren=4; } // DEFINE ULTIMO RENGLON

        key.i[1]=0; key.i=bioskey(0); // LEE TECLA PULSADA

        switch(key.ch[0]) {
            case '\r': // RETURN
                cadena[indice]=0x00; // TERMINA CADENA
                break;
            case '\b': // BACKSPACE
                if(indice==0) // VIGILA PRIMER CARACTER
                    gotoxy(--col,ren); cprintf(" "); // BORRA CARACTER ANTERIOR
                    gotoxy(col,ren); indice--; // ACTUALIZA INDICE
                break;
            case ESC : // ESCAPE
                key.ch[0] = '\r'; // DEFINE ENTER PARA SALIR
                cadena[0]=0x00; // VACIA CADENA
                break;
            default: // AÑADE CARACTER A CADENA
                if( (tipo[0]!='g') && (key.ch[0]!='.') && // DATO PARA GPS
                    (key.ch[0]!='S') && (key.ch[0]!='.') &&
                    (key.ch[0]!='.') && (isalnum(key.ch[0])!=0) ) {
                    sound(1000); delay(100); nosound(); break; }
                if( (tipo[0]!='d') && // DATO NUMERICO
                    (isdigit(key.ch[0])!=0) && (key.ch[0]!='.') ) {
                    sound(1000); delay(100); nosound(); break; }
                if( (tipo[0]!='a') && (isalnum(key.ch[0])!=0) ) // DATO ALFANUMERICO
                    sound(1000); delay(100); nosound(); break; }

                cadena[indice++]=key.ch[0]; // AÑADE CARACTER
                gotoxy(col++,ren); cprintf("%c",key.ch[0]); // DESPLIEGA CARACTER
                break;
        }
    } while((key.ch[0]!='\r') && (indice<=size));

    cadena[indice]=0x00; // ASEGURA FIN DE CADENA

    if(col!=1) { ren++; col=1; } // VIGILA FIN DE RENGLON
    if(ren> 4) {                // VIGILA FIN DE VENTANA
        gotoxy( 1, 1); delline(); // BORRA PRIMER RENGLON DE LA VENTANA
        scroll( 2,18,39,18,BLUE); // LIMPIA ULTIMO RENGLON DE LA VENTANA
        col=1; ren=4; } // DEFINE ULTIMO RENGLON

    window( 1, 1,80,25); // VUELVE A PANTALLA COMPLETA
    return;
} // TERMINA LEEDATO

/* MANEJA LA APERTURA DE ARCHIVOS DEL SISTEMA
FILE, NOMBRE DEL ARCHIVO DE DATOS
PFFILE, APUNTAADOR LOGICO AL ARCHIVO DE DATOS
*/

```

```

void abrefile(char file[12], FILE **pfile)
{
    struct ftime ft;

    if(file[0] != 0x00) { // REVISAR QUE NO HAYA ARCHIVOS ABIERTOS
        comenta("Ya existe un archivo abierto.",1); return ; }

    comenta("Archivo [8 caracteres]: ",0); // SOLICITA ARCHIVO DE DATOS
    leedato(file,8,"alnum");

    if(file[0]==0x00) return; // VIGILA QUE HAYA ARCHIVO

    strcpy(file+(strlen(file)),".BIN"); // AÑADE EXTENSION DEL ARCHIVO

    switch(MODO) {
        case VTR: // ARCHIVO PARA MODO TIERRA
            if(access(file,0)==0) { // VERIFICA ACCESO LIBRE AL ARCHIVO
                *pfile=fopen(file,"rb"); // ABRE ARCHIVO DE DATOS PARA LEER
                comenta("Archivo abierto ... ",0); comenta(file,1);
                getftime(fileno(*pfile),&ft); // LEE FECHA DE ARCHIVO
                scroll(42, 3,59, 3,BLUE); gotoxy(45, 3);
                cprintf("%u/%u/%u",ft.ft_day,ft.ft_month,ft.ft_year+1980); }
            else {
                comenta("Archivo no encontrado ...",1); file[0]=0x00; }
            break;
        case CAM: // ARCHIVO PARA MODO VUELO
            if(access(file,0)==0) { // VERIFICA ACCESO LIBRE AL ARCHIVO
                comenta(file,1); // EL ARCHIVO YA EXISTE
                comenta("Añade archivo [S/M/? ",1);
                window( 2,15,39,18); // SALTA A VENTANA DE COMENTARIOS
                gotoxy(col,ren); // SALTA A ULTIMA POSICION DEL CURSOR
                key.i=0;
                while(key.ch[0]!='s' && key.ch[0]!='S' &&
                    key.ch[0]!='M' && key.ch[0]!='m' &&
                    key.ch[0]!='ESC') key.i=bioskey(0);
                window( 1, 1,80,25); // VUELVE A PANTALLA COMPLETA
                if(key.ch[0]!='S') { key.ch[0]='s'; }
                *pfile=fopen(file,"wb"); // ABRE ARCHIVO DE DATOS
                comenta("Archivo abierto ... ",1); }
            else {
                comenta("Ningun archivo abierto ...",1); file[0]=0x00; }
            }
        else {
            *pfile=fopen(file,"wb"); // CREA ARCHIVO DE DATOS
            comenta("Archivo creado ... ",0); comenta(file,1); }
            break;
        default:
            return ;
    } // TERMINA ABREFILES

    /* MANEJA EL CIERRE DEL ARCHIVO DE DATOS
    FILE, NOMBRE DEL ARCHIVO
    PFILE, APUNTAADOR LOGICO AL ARCHIVO */
    void closefile(char file[12], FILE **pfile)
    {
        if(file[0] != 0x00) { // VERIFICA QUE EXISTA EL ARCHIVO
            fclose(*pfile); comenta("Archivo cerrado ... ",0);
            comenta(file,1); file[0]=0x00; }
        else { // EL ARCHIVO NO EXISTE
            sound(100); delay(100); nosound();
            comenta("Ningun archivo abierto.",1); }
        return ;
    }
}

```

```

} // TERMINA CERRAR

/* MANEJA LAS FUNCIONES CONTROL DE LAS CAMARAS DE VIDEO
ORDEN, FUNCION A REALIZAR POR LAS CAMARAS DE VIDEO
void ctrlcamera(unsigned char orden)
{
    long cuadro;

    comenta("Ejecutando comando ...",0);
    switch(orden) {
        case Rec : // GRABAR IMAGENES
            if((fcam[0x00].code==Rec) | // CAMARAS A STBY SI ESTAN EN REC
                for(camara=0; camara<CAMARAS-1; camara++) {
                    fcam[camara].code=Stby; // ACTUALIZA FUNCION
                    fcam[camara].name=" STBY ";
                    outportb(Wcamara,camara); delay(250); // ELIGE NO. DE CAMARA
                    outportb(Control,orden); delay(700); // ENVIA FUNCION
                    outportb(0x21,0xB0); // DISABLE IRQ7 (SINC.NOIMG)
                } else { // CAMARAS A REC SI ESTAN EN STBY
                    for(camara=0; camara<CAMARAS-1; camara++) {
                        fcam[camara].code=Rec; // ACTUALIZA FUNCION
                        fcam[camara].name=" REC ";
                        outportb(Wcamara,camara); delay(250); // ELIGE NO. DE CAMARA
                        outportb(Control,orden); delay(600); // ENVIA FUNCION
                        outportb(0x21,0x30); // ENABLE IRQ7 (SINC.NOIMG)
                    }
                } break;

            case ZoomIn: // ACERCAR IMAGEN
                outportb(Wcamara,camara); delay(250); // ELIGE NO. DE CAMARA
                outportb(Control,orden); // ENVIA FUNCION
                cuadro=nocuadro;
                while(!kbhit())
                    if(nocuadro!=cuadro) almacena(); // ACTUALIZA DATOS (SENSORES,GPS)
                break;

            case ZoomOut: // ALEJAR IMAGEN
                outportb(Wcamara,camara); delay(250); // ELIGE NO. DE CAMARA
                outportb(Control,orden); // ENVIA FUNCION
                cuadro=nocuadro;
                while(!kbhit())
                    if(nocuadro!=cuadro) almacena(); // ACTUALIZA DATOS (SENSORES,GPS)
                break;

            case Fast : // AVANZA CINTAS
                if((fcam[0x00].code==Fast) | // SALE SI ES ORDEN ACTUAL
                    if((fcam[0x00].code==Stop) ctrlcamera(Stop); // DETIENE CINTA PRIMERO
                    for(camara=0; camara<CAMARAS-1; camara++) {
                        fcam[camara].code=Fast; // ACTUALIZA FUNCION
                        fcam[camara].name=" FF ";
                        outportb(Wcamara,camara); delay(250); // ELIGE NO. DE CAMARA
                        outportb(Control,orden); delay(400); // ENVIA FUNCION
                    }
                } else { sound(1000); delay(100); nosound(); }
                break;

            case Stop : // DETIENE CINTAS
                if((fcam[0x00].code==Stop) | // SALE SI ES FUNCION ACTUAL
                    for(camara=0; camara<CAMARAS-1; camara++) {
                        fcam[camara].code=Stop; // ACTUALIZA FUNCION
                        fcam[camara].name=" STOP ";
                        outportb(Wcamara,camara); delay(250); // ELIGE NO. DE CAMARA
                        outportb(Control,orden); delay(400); // ENVIA FUNCION
                    }
                } else { sound(1000); delay(100); nosound(); }
                break;

            case Rew : // REGRESA CINTAS
                if((fcam[0x00].code==Rew) | // SALE SI ES FUNCION ACTUAL

```

```

                    if((fcam[0x00].code==Stop) ctrlcamera(Stop); // PRIMERO DETIENE CINTAS
                    for(camara=0; camara<CAMARAS-1; camara++) {
                        fcam[camara].code=Rew; // ACTUALIZA FUNCION
                        fcam[camara].name=" REW ";
                        outportb(Wcamara,camara); delay(250); // ELIGE NO. DE CAMARA
                        outportb(Control,orden); delay(400); // ENVIA FUNCION
                    }
                } else { sound(1000); delay(100); nosound(); }
                break;

            case Play : // REPRODUCIR
                if((fcam[0x00].code==Play) | // SALE SI ES ORDEN ACTUAL
                    if((fcam[0x00].code==Stop) ctrlcamera(Stop); // DETIENE CINTAS PRIMERO
                    for(camara=0; camara<CAMARAS-1; camara++) {
                        fcam[camara].code=Play; // ACTUALIZA FUNCION
                        fcam[camara].name=" PLAY ";
                        outportb(Wcamara,camara); delay(250); // ELIGE NO. DE CAMARA
                        outportb(Control,orden); delay(400); // ENVIA FUNCION
                    }
                } else { sound(1000); delay(100); nosound(); }
                break;

            case Pausa : // CINTAS EN PAUSA
                if((fcam[0x00].code==Play) | // SALE SI ES INVALIDO
                    ((fcam[0x00].code==Pausa) |
                    for(camara=0; camara<CAMARAS-1; camara++) {
                        if((fcam[camara].code==Pausa) |
                            fcam[camara].code=Play; // ACTUALIZA FUNCION
                            fcam[camara].name=" PLAY "; |
                        } else {
                            fcam[camara].code=Pausa; // ACTUALIZA FUNCION
                            fcam[camara].name=" PAUSA "; |
                            outportb(Wcamara,camara); delay(250); // ELIGE NO. DE CAMARA
                            outportb(Control,orden); delay(400); // ENVIA FUNCION
                        }
                    } else { sound(1000); delay(100); nosound(); }
                } break;

            default:
                for(camara=0; camara<CAMARAS-1; camara++) { // ACTUALIZA DATOS EN PANTALLA
                    gotoxy(42+(camara*9),15); cprintf("%8s",fcam[camara].name); }

                comenta(" Terminado",1);
                delay(1); outportb(Control,0x00); // LIMPIA REG. DE CONTROL REMOTO
                delay(1); outportb(Wcamara,0x00); // LIMPIA REG. DE NO. DE CAMARA

                return ;
            } // TERMINA CTRLCAMARA

/* MANEJA EL DESPLIEGUE DE DATOS EN PANTALLA */
void displaydatos(unsigned char tipo)
{
    char *status[8]={ "NO RESPONDE"," OCIOSO ",
                    " BUSQUEDA "," ALMANAQUE ",
                    " EFEMÉRIDES "," ADQUISICIÓN",
                    " POSICIÓN "," NAVEGACIÓN " }; // ESTADOS DE OP. DEL GPS

    char *hora; hora=malloc(8); *hora=0x00; // PARA MANEJAR LA HORA

    // DESPLIEGA ESTADO DE OPERACION DE LOS SATELITES Y DEL GPS
    if(((tipo==CAM) | (tipo==REC) | (tipo==GPS)) {
        scroll(42, 3,59, 3,BLUE); gotoxy(47, 3); cprintf("%8s",vgps.fecha);
        scroll( 2, 3,24, 4,BLUE); // LIMPIA VENTANA DE SATELITES
        for(indice=0; indice<4; indice++) {
            gotoxy(3+(indice*4), 3); cprintf("%2d",vgps.numsat[indice]);

```

ESTA TESIS NO DEBE
 SALIR DE LA BIBLIOTECA

```

gotoxy(3+(indice*4), 4); cprintf("82s",vgps.edosat[indice]);
}
scroll(27, 3,39, 3,BLUE); // LIMPIA VENTANA DE STATUS GPS
gotoxy(28, 3); cprintf("8s",status[vgps.status]);
}
if(tipo=="VTR") { // DA FORMATO DE SALIDA AL CAMPO DE HORA
*vgps.hora=0x00;
if(reg.hora==0x00) strcpy(hora,"000000"); else itoa(reg.hora,hora,10);
strncat(vgps.hora,(hora+0),2); *(vgps.hora+2)=0x00; strcat(vgps.hora,".");
strncat(vgps.hora,(hora+2),2); *(vgps.hora+5)=0x00; strcat(vgps.hora,".");
strncat(vgps.hora,(hora+4),2); *(vgps.hora+8)=0x00;
}
// DESPLIEGA HORA, LATITUD, LONGITUD, ALTITUD Y ERROR ESTIMADO EN PANTALLA
scroll(62, 3,79, 3,BLUE); gotoxy(67, 3); cprintf("8s", vgps.hora);
scroll(42, 6,59, 6,BLUE); gotoxy(47, 6); cprintf("88.2f",reg.latitud);
scroll(62, 6,79, 6,BLUE); gotoxy(67, 6); cprintf("88.2f",reg.longitud);
scroll(42, 9,59, 9,BLUE); gotoxy(47, 9); cprintf("88ld", reg.altitud);
scroll(62, 9,79, 9,BLUE); gotoxy(67, 9); cprintf("88d", reg.error);

// DESPLIEGA NO. DE IMAGEN DE CADA CAMARA EN PANTALLA
for(indice=0; indice<CAMARAS-1; indice++) {
gotoxy(42+(indice*9),18); cprintf(" 88ld",imgcam[indice]); }

// DESPLIEGA DATOS ENTREGADOS POR LOS SENSORES EN PANTALLA
for(indice=0; indice<SENSORES-1; indice++) {
if(indice<= 3) {
gotoxy( 3+(indice*9),21); cprintf("88.2f",vsensor[indice]); }
if((indice>= 4)&&(indice<= 7)) {
gotoxy(43+((indice- 4)*9),21); cprintf("88.2f",vsensor[indice]); }
if((indice>= 8)&&(indice<=11)) {
gotoxy( 3+((indice- 8)*9),23); cprintf("88.2f",vsensor[indice]); }
if((indice>=12)&&(indice<=15)) {
gotoxy(43+((indice-12)*9),23); cprintf("88.2f",vsensor[indice]); }
}
return ;
} // TERMINA DISPLAYDATOS

/* INICIALIZA VARIABLES DE LOS MODOS DE OPERACION DEL SISTEMA
MODO, MODO DE OPERACION DEL SISTEMA */
void startvar(unsigned char modo)
{
// INICIALIZA ESTADO Y NO. DE IMAGEN DE LAS CAMARAS, UNA POR UNA
for(indice=0; indice<3; indice++) {
fcam[indice].code=Off; // ESTADO DE OPERACION EN OFF
fcam[indice].name=" OFF "; //
imgcam[indice]=0x00; } // NO. DE IMAGEN EN CEROS

// DESPLIEGA EN PANTALLA ESTADO INICIAL DE LAS CAMARAS
scroll(42,15,79,15,BLUE); // LIMPIA VENTANA DE CAMARAS
for(indice=0; indice<3; indice++) {
gotoxy(42+(indice*9),15); cprintf("88s",fcam[indice].name); }

// DESPLIEGA ESTADO INICIAL DE LOS SENSORES
scroll(2,21,39,21,BLUE); scroll(42,21,79,21,BLUE);
scroll(2,23,39,23,BLUE); scroll(42,23,79,23,BLUE);
for(indice=0; indice<15; indice++) {
vsensor[indice]=0;
if(indice<= 3) {
gotoxy( 3+(indice*9),21); cprintf(" OFF "); }
if((indice>= 4)&&(indice<= 7)) {
gotoxy(43+((indice- 4)*9),21); cprintf(" OFF "); }
if((indice>= 8)&&(indice<=11)) {
gotoxy( 3+((indice- 8)*9),23); cprintf(" OFF "); }
}
}

```

```

if((indice>=12)&&(indice<=15)) {
gotoxy(43+((indice-12)*9),23); cprintf(" OFF "); }
}

file=malloc(12); *file=0x00; // LIMPIA VARIABLES DEL ARCHIVOS

reg.latitud=0x00; reg.longitud=0x00; // LIMPIA REGISTRO DE DATOS
reg.altitud=0x00; reg.hora =0x00; // LATITUD, LONGITUD, ALTITUD,
reg.error =0x00; reg.noimagen=0x00; // HORA, ERROR, NO. IMAGEN,
reg.alabeo =0x00; reg.cabeceo =0x00; // ALABEO, CABECEO, BRUJULA
reg.brujula=0x00;

ren =1; col =1; // COORDENADAS VENTANA DE COMENTARIOS
renfun=1; colfun=1; // COORDENADAS VENTANA DE FUNCIONES

nocuadro=0x00; // NO. DE IMAGEN LEIDO DE LAS CAMARAS

// LIMPIA HORA Y FECHA
vgps.hora =malloc( 8); strcpy(vgps.hora, "00:00:00");
vgps.fecha=malloc(12); strcpy(vgps.fecha,"00/00/0000");

if(modo=="VTR") { // LIMPIA VARIABLES EXCLUSIVAS DEL MODO TIERRA
consulta=0x00; // BANDERA DE COMUTACION DE CAMARAS
camara =0x00;
}
if(modo=="CAM") { // LIMPIA VARIABLES EXCLUSIVAS DEL MODO VUELO
for(indice=0; indice<=4; indice++) { // LIMPIA VARIABLES DE GPS
vgps.numset[indice]=0x00; // NUMERO DE SATELITE
vgps.edosat[indice]=0x00; } // ESTADO DE SATELITE

vgps.status=0x00; // STATUS DEL GPS
vgps.curso =0x00; vgps.speed =0x00; // CURSO Y VELOCIDAD

byte1=0x00; byte2=0x00; byte3=0x00; // BYTES DE NO. IMAGEN
pulsio=0x01; nobyteimg=0x00; // PULSO Y NO. BYTE DE NO. IMAGEN

newreggps =0x00; // NUEVO REGISTRO-GPS RECIBIDO
nomsggps =0x01; chrpgps =0x00; // NO. MSG. GPS, CARACTER MSG. GPS
msggps1[0]=0x00; msggps2[0]=0x00; // MENSAJES RECIBIDOS DE GPS

despierta=0x00; // NO. IMAGEN, DESPIERTA CAMARAS
}
return;
} // TERMINA STARTVAR

/*
-----
TERMINAN FUNCIONES GENERALES DEL SISTEMA
-----*/

/*
-----
FUNCIONES EXCLUSIVAS DEL MODO DE OPERACION EN TIERRA
-----*/

/*
-----
===== REDEFINE DE LOS SERVICIOS DE INTERRUPCION DE LA COMPUTADORA =====*/

/*
REDEFINE SERVICIO DE INTERRUPCION IRQ4 (DEL PUERTO SERIE 1) PARA LEER LA
INFORMACION DE CADA IMAGEN (DOCE BYTES: UN BYTE DE CONTROL Y ONCE BYTES
DE DATOS)
*/
void interrupt leenoimagen(void)
{
disable(); // BLOQUEA INTERRUPCIONES

if(((byte1=inporth(0x3fd))&0x1E)!=0x00) { // ELIMINA ERROR SI EXISTE

```

```

byte1=inportb(0x3f8); nbyteimg=0x00; }
else { // LEE BYTES DE INFORMACION DE CADA IMAGEN
switch(nbyteimg) {
case 0: if(inportb(0x03f8)==0x80) nbyteimg=0x01; // BYTE DE CONTROL
break;
case 1: byte1=inportb(0x03f8); // BYTE 1
if(byte1 < 0x80) nbyteimg=0x02;
break;
case 2: byte2=inportb(0x03f8); // BYTE 2
if(byte2 < 0x80) nbyteimg=0x03;
break;
case 3: byte3=inportb(0x03f8); // BYTE 3
if(byte1 < 0x80) nbyteimg=0x04;
break;
case 4: byte4=inportb(0x03f8); // BYTE 4
if(byte2 < 0x80) nbyteimg=0x05;
break;
case 5: byte5=inportb(0x03f8); // BYTE 5
if(byte1 < 0x80) nbyteimg=0x06;
break;
case 6: byte6=inportb(0x03f8); // BYTE 6
if(byte2 < 0x80) nbyteimg=0x07;
break;
case 7: byte7=inportb(0x03f8); // BYTE 7
if(byte1 < 0x80) nbyteimg=0x08;
break;
case 8: byte8=inportb(0x03f8); // BYTE 11
nocuadro=(byte3*0x4000)+(byte2*0x80)+byte1; // PROCESA INFORMACION
imgcam[camara]=nocuadro; // ACTUALIZA NO.IMAGEN
if(conmuta!=0x00) {
if(camara<=(CAMARAS-1)) camara+=0xFC; // A CAMARA CERO
else camara++; // SIGUIENTE CAMARA
outportb(Ncamara,camara); // ELIGE CAMARA
conmuta=0x00;
}
break;
default: ;
}
} // TERMINA ELSE

outportb(0x20, 0x64); // ANULA IRQ4 : PUERTO SERIE 1
enable(); // PERMITE INTERRUPCIONES
} // TERMINA LEEIMAGEN

/* REDEFINE SERVICIO DE INTERRUPCION IRQ7 (PUERTO PARALELO 1) PARA SENSAR
LOS PULSOS DE SINCRONIA VERTICAL DE LA CAMARA 1 Y PODER SALTAR A OTRA
CAMARA SI DESPUES DE 5 PULSOS NO SE RECIBE INFORMACION DE SU CANAL DE
AUDIO
void interrupt sincronia(void)
{
disable(); // BLOQUEA INTERRUPCIONES
outportb(0x20, 0x67); // ANULA IRQ7 : PUERTO PARALELO 1

if(++conmuta>=5) { // REvisa 5 PULSOS O MAS DE SINCRONIA
if(camara>=(CAMARAS-1)) camara+=0xFC; // A CANAL CERO (CAMARA 1)
else camara++; // SIGUIENTE CAMARA
outportb(Ncamara,camara); // ELIGE CAMARA
conmuta=0x00;
nbyteimg=0x00;
}
enable(); // PERMITE INTERRUPCIONES
} // TERMINA SINCRONIA

```

```

/* TERMINA REDEFINICION DE LOS SERVICIOS DE INTERRUPCION DE LA COMPUTADORA -
-----*/
/* ----- FUNCIONES DE PROCESAMIENTO DEL ARCHIVO DE DATOS -----*/
/* GENERA ARCHIVO ASCII DE DATOS CON EL MISMO NOMBRE DEL ARCHIVO ORIGINAL
PERO CON EXTENSION TXT
FILEREg, NOMBRE DEL ARCHIVO DE DATOS
PFILEReg, APUNTADOR LOGICO DEL ARCHIVO DE DATOS
void asciifile(char filereg[12], FILE **pfilereg)
{
char *punto;
char *file: FILE *pfile; // VARIABLES PARA ARCHIVO ASCII

if(filereg[0]==0x00) { // REvisa QUE HAYA ARCHIVO ABIERTO
sound(1000); delay(100); nosound();
comenta("Ningún archivo abierto ...",1);
return; }

// INICIA GENERACION DE ARCHIVO ASCII DE DATOS
comenta("Creando archivo ASCII ...",1); file=malloc(12);

// CREA EL ARCHIVO ASCII
strcpy(file,filereg); punto=strchr(file,"."); // CREA NOMBRE
file[punto-file]=0x00; strcpy(file+(strlen(file)),".TXT");
pfile=fopen(file,"wt"); // ABRE ARCHIVO

fseek(*pfilereg, 0,SEEK_SET); // SALTA AL INICIO DEL ARCHIVO BINARIO

// ESCRIBE ENCABEZADO DEL ARCHIVO
fprintf(pfile,"No.Im gen,Hora ,Latitud ,Longitud ,Altitud ,");
fprintf(pfile,"Error ,Alabeo ,Cabeceo,Brujula\n");

// ESCRIBE EL ARCHIVO DE DATOS ASCII
while(!feof(*pfilereg)) { // NASTA EL FIN DE ARCHIVO BINARIO
fread(&reg,wordreg,1,*pfilereg); // LEE REGISTRO DE ARCHIVO BINARIO
// ESCRIBE LINEA DE ARCHIVO ASCII
fprintf(pfile, "%19d,%19d,%9.2f,%9.2f,%9d,%9d,%7u,%7u\n",
reg.noimagen, reg.hora, reg.latitud, reg.longitud, reg.altitud,
reg.error, reg.alabeo, reg.cabeceo, reg.brujula);
}
fclose(pfile); // CIERRA ARCHIVO ASCII

// FINALIZA ARCHIVO ASCII
comenta("Archivo terminado ...",0); comenta(file,1);

free(file);
return;
} // TERMINA ASCIIFILE

/* BUSCA INFORMACION EN EL ARCHIVO DE DATOS UTILIZANDO EL METODO DE
BISECCION, YA QUE LOS CAMPOS DE DATOS DE BUSQUEDA ESTAN EN UN ORDEN
ASCENDENTE
TIPO, CAMPO SOBRE EL QUE SE REALIZARA LA BUSQUEDA
VALOR, VALOR DEL CAMPO A LOCALIZAR
PFILE, APUNTADOR LOGICO AL ARCHIVO SOBRE EL QUE SE BUSCARA EL DATO
void finddato(char tipo[7], long valor, FILE **pfile)
{
long regsup, reginf; ldiv_t mitad;
char *campo; campo=malloc(10); *campo=0x00;

```

```

// SOLICITA DATO DE LA IMAGEN
if(valor==0) {
    if(tipo[0]=='i') comenta("No. de Imagen: ",0); // SI NO SE DA EL VALOR SOLICITA DATO
    else comenta("Hora [hh:mm:ss]: ",0); // NO. DE LA IMAGEN
    leedato(campo,10,"digit"); // HORA DE LA IMAGEN
    // LEE DATO DE TECLADO

    if((*campo==0x00) || (valor=atoi(campo))) return; // REvisa que haya dato

// REvisa que el archivo no este vacio
if(filelength(fileno(*pfile))==0) {
    comenta("Archivo Vacio ... ",1); return; // ARCHIVO VACIO

// INICIA BUQUEDA POR EL METODO DE LA BISECCION
comenta("Busca Información ... ",0);
reginf=0; // LIMITE INFERIOR
regsup=(filelength(fileno(*pfile))/wordreg)-1; // LIMITE SUPERIOR

while(reginf<regsup) { // HASTA ENCONTRAR VALOR
    mitad=ldiv((reginf+regsup),2); // CALCULA REGISTRO MEDIO
    fseek(*pfile,(mitad.quot-1)*wordreg,SEEK_SET); // SALTA AL REGISTRO MEDIO
    fread(&reg,wordreg,1,*pfile); // LEE REGISTRO
    if(tipo[0]=='i') { // ESTABLECE NUEVOS LIMITES
        if(reg.noimagen<valor) reginf=mitad.quot+1; // NUEVO LIMITE INFERIOR
        else regsup=mitad.quot; // NUEVO LIMITE SUPERIOR
    } else {
        if(reg.hora<valor) reginf=mitad.quot+1; // NUEVO LIMITE INFERIOR
        else regsup=mitad.quot; // NUEVO LIMITE SUPERIOR
    }

    mitad=ldiv((reginf+regsup),2); // CALCULA REGISTRO FINAL
    fseek(*pfile,mitad.quot-1)*wordreg,SEEK_SET); // SALTA AL REGISTRO FINAL
    fread(&reg,wordreg,1,*pfile); // LEE REGISTRO

    comenta("Terminado",1);
    free(campo);

    gotonoimg(reg.noimagen); // BUSCA NO. IMAGEN EN LAS CAMARAS
    displaydatos(MODO); // ACTUALIZA DATOS EN PANTALLA
    return;
} // TERMINA FINDDATO

/* BUSCA NUMERO DE IMAGEN INDICADO EN LAS CAMARAS DE VIDEO
IMAGEN, NUMERO DE IMAGEN A BUSCAR
void gotonoimg(long imagen)
{
    long tiempo=0; // PARA CALCULAR TIEMPO DE APROXIMACION
    char *num; num=malloc(8); *num=0x00;

// INICIA BUSQUEDA DE IMAGEN
comenta("Busca Videoimagen ... ",0);

tiempo=ceil(fabs(nocuadro-imagen)/0.6); // CALCULA TIEMPO DE APROXIMACION

// APROXIMACION RAPIDA A LA IMAGEN INDICADA
if((nocuadro-imagen)<0) { ctrlcamara(Rew); tiempo+=100; } // REGRESA CINTAS
else { ctrlcamara(Fast); tiempo-=100; } // AVANZA CINTAS
delay(tiempo); // TIEMPO DE APROXIMACION
ctrlcamara(Stop); // DETIENE CINTAS

// APROXIMACION LENTA A LA IMAGEN INDICADA
ctrlcamara(Play); // REPRODUCE CINTAS
while(nocuadro<imagen); // AVANZA HASTA MO. IMAGEN
ctrlcamara(Pausa); // CINTAS EN PAUSA

```

```

// SIMCRONIZA IMAGENES DE TODAS LAS CAMARAS
for(indice=0; indice<CAMARAS-1; indice++)
    if(imagen<imgcam[indice]) imagen=imgcam[indice];

outportb(0x21,0xA8); // BLOQUEA IRQ7: EVITA CONMUTACION DE CAMARAS

for(indice=0x00; indice<CAMARAS-1; indice+=0x05)
    if(imagen<imgcam[indice]) {
        nobyteimg=0x00; conmuta=0x00;
        outportb(Ncamara,indice); // ELIGE CAMARA
        outportb(Control,Pausa); // CAMARA A PLAY
        while(imagen<imgcam[indice]); // AVANZA HASTA MO. IMAGEN
        outportb(Ncamara,indice); // ELIGE CAMARA
        outportb(Control,Pausa); // CAMARA A PAUSA
    }
outportb(0x21,0x28); // RESTAURA IRQ7:PERMITE CONMUTACION DE CAMARAS

// TERMINA SIMCRONIZACION
comenta("Terminado",1);
free(num);
return;
} // TERMINA GOTONOIMG

/* ===== TERMINAN FUNCIONES DE PROCESAMIENTO DEL ARCHIVO DE DATOS =====
===== */

/* INICIALIZA Y CONTROLA LAS FUNCIONES DEL MODO DE OPERACION DE TIERRA */
void vtr(void) { // MODO VTR: CONDICIONES INICIALES Y TAREA GENERAL

// AYUDA EN LINEA DE CADA FUNCION DE CONTROL
char *onlinevtr[10]={
    "Termina Mode Tierra", "Adelanta Cintas de Video",
    "Detiene Cintas de Video", "Regresa Cintas de Video",
    "Reproduce Cintas de Video", "Cintas de Video en Pausa",
    "Convierte Archivos a formato ASCII", "Salta a Imagen Número ...",
    "Salta a Imagen con Hora ... "};

long cuadro; // PARA EL NUMERO DE IMAGEN DE LA CAMARA

void interrupt leenoimagen(); // NUEVOS SERVICIOS DE INTERRUPCION
void interrupt sincronia(); // DEFINIDOS, IRQ4 E IRQ7

void interrupt (*oldint_0C)(); // APUNTADOR A SERVICIO DE INTERRUPCION
void interrupt (*oldint_0F)(); // ANTERIORES

startvar(VTR); // INICIALIZA VARIABLES

outportb(Control,0x00); // LIMPIA REG. DE CONTROL REMOTO
outportb(Ncamara,0x00); // LIMPIA REG. DE CAMARA

MODO=VTR;

scroll( 2,15,39,18,BLUE); // LIMPIA VENTANA DE COMENTARIOS

abrefile[file,spfile); // ABRE ARCHIVO DE DATOS

if(file[0]==0x00) { // REvisa que haya archivo de datos
    comenta("SE REQUIERE ARCHIVO DE DATOS",1);
    sound(1000); delay(400); nosound(); delay(3000); return; }

oldint_0C=getvect(0x0C); // GUARDA INTERRUPCIONES ORIGINALES
oldint_0F=getvect(0x0F);

```

```

setvect(0x0c,leenoimagen): // ESTABLECE NUEVOS SERVICIOS DE INTERRUPCION
setvect(0x0f,sincronia):

outportb(0x21,0x28): // HABILITA IRQ4 e IRQ7

puertoserie(1,2400,2,2): // CONFIGURA COM1 PARA RECIBIR NUMERO DE IMAGEN

displaydatos[MODO]: // DESPLIEGA DATOS EN PANTALLA
ctrlcamara(Stop): // DETIENE CAMARAS

// DESPLIEGA MENU DE FUNCIONES
scroll(2,3,39,9,BLUE); // LIMPIA VENTANA DE FUNCIONES
print(3,3,"1. CAMARA: FAST 6. PROCESA ARCHIVO",LIGHTGREEN,BLUE);
print(3,4,"2. CAMARA: STOP 7. IMAGEN NUM...",LIGHTGREEN,BLUE);
print(3,5,"3. CAMARA: REW 8. IMAGEN HORA...",LIGHTGREEN,BLUE);
print(3,6,"4. CAMARA: PLAY ",LIGHTGREEN,BLUE);
print(3,7,"5. CAMARA: PAUSA 0. SALIR ",LIGHTGREEN,BLUE);

// DESTELLEA PRIMERA OPCION Y DESPLIEGA AYUDA EN LINEA CORRESPONDIENTE
colfun=3; renfun=3; blinking(1,colfun,renfun,16,6opcion);
print(3,9,onlinevtr(),LIGHTCYAN,BLUE);

do { // LEE FUNCION ELEGIDA DEL MENU
key.i=0; gotoxy(colfun,renfun);
while(key.ch[0]!='ESC' && key.ch[0]!='\r' &&
key.i!=0x4800 && key.i!=0x4b00 &&
key.i!=0x4d00 && key.i!=0x5000) { // NAVEGA POR EL MENU DE FUNCIONES
while(!kbhit())
if(cuadro=imgcam[0]) { // SI CAMBIA NO. IMAGEN DE LAS CAMARAS
cuadro=imgcam[0];
findnoimg(imgcam[0]); // LEE DATOS ASOCIADOS DEL ARCHIVO
displaydatos[MODO]: 1 // DESPLIEGA DATOS EN PANTALLA
key.i=bioskey(0); // LEE TECLADO
}

// SI SE DIGITAN LAS TECLAS DE FLECHAS
if(!key.ch[0]!='\r' && (key.ch[0]!='ESC')) {
blinking(0,colfun,renfun,16,6opcion); // ELIMINA DESTELLO DE OPCION
// ANTERIOR
// MUEVE CURSOR POR EL MENU DE FUNCIONES
switch(key.i) {
case 0x4800: if(renfun==3) renfun=7; else renfun--; break; // UP
case 0x4b00: if(colfun==3) colfun=22; else colfun=3; break; // LEFT
case 0x4d00: if(colfun==3) colfun=22; else colfun=3; break; // RIGHT
case 0x5000: if(renfun==7) renfun=3; else renfun++; break; // DOWN
default:; }

// GENERA DESTELLO DE LA NUEVA OPCION Y DESPLIEGA SU AYUDA EN LINEA
blinking(1,colfun,renfun,16,6opcion); // BLINKING OPCION
scroll(3,3,39,9,BLUE);
print(3,9,onlinevtr[opcion-0x30],LIGHTCYAN,BLUE);
}

// SI SE DIGITO ENTER EJECUTA FUNCION CORRESPONDIENTE
if(key.ch[0]=='\r') {
switch(opcion) {
case '1': ctrlcamara(Fast); break; // ADELANTA CINTAS
case '2': ctrlcamara(Stop); break; // DETIENE CINTAS
case '3': ctrlcamara(Rew); break; // RETRASA CINTAS
case '4': ctrlcamara(Play); break; // REPRODUCE CINTAS
case '5': ctrlcamara(Pausa); break; // CINTAS EN PAUSA
case '6': ascii(file,spfile); break; // CONVIERTE ARCHIVOS
case '7': finddato("img",0,spfile); break; // POSICIONA EN NO. IMAGEN

```

```

case '8': finddato("hora",0,spfile); break; // POSICIONA EN HORA
case '9': digitaliza(); break; // DIGITALIZA IMAGEN
case '0': MODO=OFF; break; // TERMINA MODO TIERRA
default:; }
}

// SI SE TECLEO ESCAPE
if(key.ch[0]!='ESC') MODO=OFF; // TERMINA MODO DE OPERACION VTR

} while(!MODO==VTR);

outportb(0x21,0x28): // RESTAURA INTERRUPCIONES ORIGINALES
setvect(0x0c,oldint_0C): // RESTAURA SERVICIOS DE INTERRUPCION
setvect(0x0f,oldint_0F): // ORIGINALES IRQ4 E IRQ7

puertoserie(1,9600,1,1,1): // RESTAURA PUERTO SERIE 1
puertoserie(2,9600,1,1,1): // RESTAURA PUERTO SERIE 2

closefile(file,spfile); // CIERRA ARCHIVO DE DATOS

free(file);
return;
} // TERMINA VTR

/* -----
TERMINAN FUNCIONES EXCLUSIVAS DEL MODO DE OPERACION DE TIERRA
----- */

/* -----
FUNCIONES DE LA CAMPAÑA DE DETECCION DE IMAGENES
----- */

/* -----
===== FUNCIONES DE ADQUISICION Y ALMACENAMIENTO DE DATOS DEL SISTEMA =====
----- */

/* ALMACENA DATOS GENERADOS POR EL SISTEMA
PFILE, APUNTA A LOGICO AL ARCHIVO DE DATOS
void almacena(FILE *pfile) */

float hora, altitud; float latitud, longitud; int error;

// LEE Y ACTUALIZA INFORMACION PROPORCIONADA POR LOS SENSORES
sensor[0x01,4reg.alabeo]; sensor[0x02,4reg.cabeceo];
sensor[0x03,4reg.brujula];

// ACTUALIZA NUMEROS DE IMAGEN LEIDOS DE LAS CAMARAS DE VIDEO
imgcam[0]=nocuadro; imgcam[1]=nocuadro;
imgcam[2]=nocuadro; imgcam[3]=nocuadro;

// SI EXISTE UN NUEVO MENSAJE DEL GPS
if(newmsggps) {
if(newmsggps) { // REVISA EL NUMERO DE MENSAJE DEL GPS
procesamsggps(msggps2); // PROCESA EL SEGUNDO MENSAJE DEL GPS
msggps2[0]=0x00; } // LIMPIA SEGUNDO MENSAJE DEL GPS
else {
procesamsggps(msggps1); // PROCESA EL PRIMER MENSAJE DEL GPS
msggps1[0]=0x00; } // LIMPIA PRIMER MENSAJE DEL GPS
newmsggps=0x00; // ESTABLECE QUE NO HAY NUEVO MENSAJE DEL GPS
}

// SI LAS CAMARAS ESTAN GRABANDO
if(!((cam[0].code==Rec) && (reg.noimagen==nocuadro))) {
reg.noimagen=nocuadro; // ACTUALIZA NO. DE IMAGEN
if(newreggps) { // SI EXISTE NUEVO REGISTRO DEL GPS

```

```

newreggps=0x00; // ESTABLECE QUE NO EXISTE NUEVO REGISTRO DE GPS
fwrite(&reg,wordreg,1,*pfile); // ALMACENA REGISTRO EN EL ARCHIVO
else {
// COPIA VALORES DE GPS DEL REGISTRO
hora=reg.hora; altitud=reg.altitud; latitud=reg.latitud;
longitud=reg.longitud; error=reg.error;
// LIMPIA VALORES DE GPS DEL REGISTRO
reg.hora=0; reg.altitud=0; reg.latitud=0;
reg.longitud=0; reg.error=0;

fwrite(&reg,wordreg,1,*pfile); // ALMACENA REGISTRO EN EL ARCHIVO

// RESTAURA VALORES DE GPS DEL REGISTRO
reg.hora=hora; reg.altitud=altitud; reg.latitud=latitud;
reg.longitud=longitud; reg.error=error; }
return;
// TERMINA ALMACENA

/* ADQUIERE EL VALOR ENTREGADO POR EL SENSOR INDICADO Y LO DIGITALIZA
MEDIANTE EL CONVERTIDOR ANALOGICO-DIGITAL
SEN, NUMERO DE SENSOR MUESTREADO
WORD, PALABRA ENTREGADA POR EL CONVERTIDOR DEL VALOR DEL SENSOR */
void sensor(unsigned char sen, unsigned word)
{
unsigned msb, lsb; // MSB Y LSB DE LA PALABRA DEL CA/D
sen=sen<<1; msb=0x00; lsb=0x00; // AJUSTA MO. DE SENSOR A MUESTREAR

outp(Nsensor, sen&0xf); // MUESTREA EL VALOR DEL SENSOR
outp(Nsensor, sen|0x01); // RETIENE EL VALOR DEL SENSOR

delay(1); outp(Convert(10x00,0x00); // REALIZA CONVERSION DE 12 BITS
delay(1); msb=inport(Convert(40x00,0x00)); // RECOGE 8MSB
delay(1); lsb=inport(Convert(10x0001)); // RECOGE 4LSB Y 4CEROS

*word=((msb<<4)|(lsb>>4)); *word&=0x0fff; // FORMA PALABRA DE SALIDA
return;
// TERMINA SENSOR

/* PROCESA MENSAJE ENVIADO POR EL RECEPTOR GPS EXTRAYENDO LAS VARIABLES
UTILES AL SISTEMA
MENSAJE, MENSAJE ENVIADO POR EL RECEPTOR GPS */
void procesaasggps(char mensaje[100])
{
char *campo; // VARIABLE PUENTE PARA MANEJO DE MENSAJE RECIBIDO
int indice; // PARA CICLO DE CONTEO
int tipomsg=0; // PARA ESTABLECER TIPO DE MENSAJE RECIBIDO

campo=malloc(12);

// REvisa LA VALIDEZ DEL MENSAJE RECIBIDO
if(!revisachksum(mensaje)) {

// ESTABLECE EL TIPO DE MENSAJE RECIBIDO
if(strcmp(mensaje,"$GPZDA,")==mensaje) tipomsg=GPZDA; // HORA Y FECHA
if(strcmp(mensaje,"$GPGGA,")==mensaje) tipomsg=GPGGA; // POSICION
if(strcmp(mensaje,"$PNGLB,")==mensaje) tipomsg=PNGLB; // ALTITUD
if(strcmp(mensaje,"$GPVTG,")==mensaje) tipomsg=GPVTG; // CURSO
if(strcmp(mensaje,"$PNGLF,")==mensaje) tipomsg=PNGLF; // SATELITES
if(strcmp(mensaje,"$PNGLG,")==mensaje) tipomsg=PNGLG; // ERROR
if(strcmp(mensaje,"$PNGLM,")==mensaje) tipomsg=PNGLM; // ESTADO GPS

// PROCESA EL MENSAJE SEGUN SU TIPO

```

```

switch(tipomsg) {
case GPZDA: // A00: HORA Y FECHA
// OBTIENE HORA
strncpy(campo,(mensaje+7),6); *(campo+6)=0x00;
reg.hora=atoi(campo);
// FORMA CAMPO DE HORA PARA VARIABLE DE GPS
strncpy(campo,(mensaje+7),2); *(campo+2)=0x00; strcat(campo,"");
strncat(campo,(mensaje+9),2); *(campo+5)=0x00; strcat(campo,"");
strncat(campo,(mensaje+11),2); *(campo+10)=0x00;
strncpy(vgps.hora,campo);
// OBTIENE FECHA
// FORMA CAMPO DE FECHA
strncpy(campo,(mensaje+14),2); *(campo+2)=0x00; strcat(campo,"");
strncat(campo,(mensaje+17),2); *(campo+5)=0x00; strcat(campo,"");
strncat(campo,(mensaje+20),4); *(campo+10)=0x00;
strncpy(vgps.fecha,campo);
break;

case GPGGA: // B00: POSICION
// OBTIENE LATITUD
strncpy(campo,(mensaje+14),7);
*(campo+8)=0x00; reg.latitud=atof(campo);
strncpy(campo,(mensaje+22),1); *(campo+1)=0x00;
if(campo[0]!='S') reg.latitud*=-1;

strncpy(campo,(mensaje+24),8); // OBTIENE LONGITUD
*(campo+9)=0x00; reg.longitud=atof(campo);
strncpy(campo,(mensaje+33),1); *(campo+1)=0x00;
if(campo[0]!='W') reg.longitud*=-1;
break;

case PNGLB: // B02: ALTITUD EXTENDIDA
// OBTIENE ALTITUD
strncpy(campo,(mensaje+10),6);
*(campo+6)=0x00; reg.altitud=atoi(campo);
break;

case GPVTG: // E00: CURSO Y VELOCIDAD
// OBTIENE CURSO
strncpy(campo,(mensaje+7),6);
*(campo+6)=0x00; vgps.curso=atoi(campo);
strncpy(campo,(mensaje+33),5); // OBTIENE VELOCIDAD
*(campo+5)=0x00; vgps.speed=atoi(campo);
break;

case PNGLF: // F02: SATELITES UTILIZADOS
for(indice=0; indice<4; indice++) {
strncpy(campo,(mensaje+(10+(indice*9))),2); // OBTIENE NO. SAT.
*(campo+2)=0x00; vgps.numsat[indice]=atoi(campo);
strncpy(campo,(mensaje+(15+(indice*9))),1); // OBTIENE EDO.SAT.
*(campo+1)=8x00; vgps.edosat[indice]=campo[0]; }
break;

case PNGLG: // G00: ERROR ESTIMADO
strncpy(campo,(mensaje+24),3); // OBTIENE ERROR ESTIMADO
*(campo+3)=0x00; reg.error=atoi(campo);
break;

case PNGLM: // M00: ESTADO DEL GPS
strncpy(campo,(mensaje+32),1);
*(campo+1)=0x00; vgps.status=atoi(campo);
newreggps=0x01; // ESTABLECE NUEVO REGISTRO DE GPS RECIBIDO
break;

default :;

```

```

    } // TERMINA SWITCH
  }
  else comenta(mensaje,1);

  free(campo);
  return;
} // TERMINA PROCESAMSGGPS

/* = TERMINAN FUNCIONES DE ADQUISICION Y ALMACENAMIENTO DE DATOS DEL SISTEMA =
-----*/

/* -----
----- FUNCIONES DE CONTROL DE CAMARA MOVIL -----*/

/* CONTROLA EL MOVIMIENTO DE LOS SERVO MOTORES DE LA CAMARA MOVIL
SERVO, NUMERO DE SERVO MOTOR PARA MOVER
GRADOS, DESPLAZAMIENTO QUE DEBEN REALIZAR LOS SERVO MOTORES */
void moveservo(unsigned servo, unsigned grados)
{
  outp(Noservo,grados); // ESTABLECE DESPLAZAMIENTO
  outp(Mvservo,servo); delay(2); // MUESTRA DESPLAZAMIENTO 2 MILISEGUNDOS
  outp(Mvservo,servo)0xFC); // RETIENE MOVIMIENTO
  return;
} // TERMINA MOVENOTOR

/* CONTROLA EL MECANISMO DE POSICIONAMIENTO DE LA CAMARA MOVIL DEL SISTEMA
(COL, REN), COORDENADAS DE APUNTAMIENTO PARA LA CAMARA MOVIL
ZOOM, DISTANCIA FOCAL DE LA CAMARA MOVIL */
void ctrlservo(unsigned col, unsigned ren, unsigned zoom)
{
  // CENTRO DE LA VENTANA QUE REPRESENTA LA IMAGEN TOMADA POR LA CAMARA MOVIL
  unsigned colcen=28, rencen=9;

  unsigned ref=0x7f; // REFERENCIA CERO DE LOS SERVO MOTORES
  float con=0.3515625; // MINIMA UNIDAD DE DESPLAZAMIENTO DE LOS SERVO
  unsigned mov1, mov2, mov3; // UNIDADES DE DESPLAZAMIENTO
  // RESPECTO AL CENTRO DE VENTANA
  float grad1, grad2, grad3; // DESPLAZAMIENTO EN GRADOS PARA CADA SERVO MOTOR

  // UNIDADES DE DESPLAZAMIENTO EN EL EJE X
  if(col>colcen) { // DESPLAZAMIENTO POSITIVO
    mov1=ref+((col-colcen)* 9)/con; // UNIDADES DE DESPLAZAMIENTO DEL EJE X
    grad1=(mov1-ref)*con; // DESPLAZAMIENTO EN GRADOS DEL EJE X
  } else { // DESPLAZAMIENTO NEGATIVO
    mov1=ref-((colcen-col)* 9)/con; // UNIDADES DE DESPLAZAMIENTO DEL EJE X
    grad1=(ref-mov1)*(-con); // DESPLAZAMIENTO EN GRADOS DEL EJE X
  }

  // UNIDADES DE DESPLAZAMIENTO EN EL EJE Y
  if(ren>rencen) { // DESPLAZAMIENTO POSITIVO
    mov2=ref+((ren-rencen)*12)/con; // UNIDADES DE DESPLAZAMIENTO DEL EJE Y
    grad2=(mov2-ref)*con; // DESPLAZAMIENTO EN GRADOS DEL EJE Y
  } else { // DESPLAZAMIENTO NEGATIVO
    mov2=ref-((rencen-ren)*12)/con; // UNIDADES DE DESPLAZAMIENTO DEL EJE Y
    grad2=(ref-mov2)*(-con); // DESPLAZAMIENTO EN GRADOS DEL EJE Y
  }

  // UNIDADES DE DESPLAZAMIENTO DEL AJUSTE DE DISTANCIA FOCAL
  mov3=zoom;
  // UNIDADES DE DESPLAZAMIENTO DEL ZOOM
  if(mov3>ref) // DESPLAZAMIENTO POSITIVO
    grad3=(mov3-ref)*con; // GRADOS DE DESPLAZAMIENTO DEL ZOOM
  else // DESPLAZAMIENTO NEGATIVO
    grad3=(ref-mov3)*(-con); // GRADOS DE DESPLAZAMIENTO DEL ZOOM
}

```

```

// MUEVE LA CAMARA A LA POSICION INDICADA
moveservo(ServoX,mov1);
moveservo(ServoY,mov2);
moveservo(SrZoom,mov3);

// DESPLIEGA EN PANTALLA LOS GRADOS DE DESPLAZAMIENTO DE LOS SERVO
scroll(42,12,52,12,BLUE); gotoxy(45,12); cprintf("%04.2f",grad1);
scroll(55,12,65,12,BLUE); gotoxy(58,12); cprintf("%04.2f",grad2);
scroll(68,12,79,12,BLUE); gotoxy(71,12); cprintf("%04.2f",grad3);

return;
} // TERMINA CTRLSERVO

/* -----
----- TERMINAN FUNCIONES DE CONTROL DE CAMARA MOVIL -----*/

/* REVISAS QUE LAS CAMARAS GRABEN. ESTA FUNCION SE ACTIVA CADA 4 MINUTOS
MIENTRAS LAS CAMARAS GRABAN, YA QUE LAS CAMARAS CAMBIAN SU FUNCIONAMIENTO
A STANDBY AUTOMATICAMENTE A LOS 5 MINUTOS CUANDO NO SE MODIFICA SU ESTADO
DE OPERACION, CUALQUIERA QUE ESTE SEA. EVITANDO ASI QUE LA CAMARA DEJE DE
GRABAR EN VUELO */
void enablecam(void)
{
  unsigned char status; // BANDERA DE GRABACION DE LAS CAMARAS
  unsigned indice; // PARA CICLO DE CONTEO

  status=inport(Cenable); // LEE BYTE DE ESTADO DE GRABACION DE LAS CAMARA

  // REVISAS LOS CUATRO BITS MAS SIGNIFICATIVOS, UNO POR UNO, QUE INDICAN
  // SI CADA CAMARA GRABA (UNO) O NO GRABA (CERO)
  for(indice=0x00; indice<CAMARAS-1; indice++) {
    if(status<>0x00) { // SI NO GRABA LA CAMARA SE INDICA
      fcam[indice].code=0x00; fcam[indice].name=" OFF "; }
  }
} // TERMINA ENABLECAM

/* CONTROLA FUNCIONES DEL MENU DE GRABACION */
void camrec(void)
{
  // AYUDA EN LINEA DE LAS FUNCIONES DEL MENU
  char *onlinerec[3] = { "Sale Record", "Graba/Espera" };

  MODO=REC;

  // DIBUJA VENTANA DE IMAGEN DE CAMARA MOVIL
  scroll( 2, 7,15,11,BLUE); marco(17, 5,40,13,WHITE,BLUE);
  print(19, 5,"Imagen Camera Movil",YELLOW,BLUE);
  for(renfun=6; renfun=10; renfun+=2) // DIBUJA FONDO DE VENTANA
    print(18,renfun, " ",WHITE,BLUE);
  print(18,renfun, " ",WHITE,BLUE);
  print(18,12, " ",WHITE,BLUE);

  // DIBUJA MENU DE FUNCIONES
  marco( 1, 6,16,12,WHITE,BLUE); print( 3, 6,"Controles",YELLOW,BLUE);
  print( 3, 7,"1. REC/STBY",LIGHTGREEN,BLUE);
  print( 3, 8,"0. SALIR ",LIGHTGREEN,BLUE);
  print( 3,10," tab ->",WHITE,BLUE);

  // DESTELLEA LA PRIMERA OPCION DEL MENU DE FUNCIONES
  colfun=3; renfun=7;
  blinking( 1,colfun,renfun,10,4opcion); // DESTELLEA OPCION
  print( 3,11,onlinerec[1],LIGHTCYAN,BLUE); // DESPLIEGA AYUDA EN LINEA
}

```

```

// MANTIENE MENU DE GRABACION HASTA ELEGIR SALIR O ESCAPE
do {
key.i=0;
while(key.ch[0]!=ESC&&key.ch[0]!='\r'&&
key.ch[0]!='\t'&&key.i!=0x4800&&
key.i!=0x4b00&&key.i!=0x4d00&&key.i!=0x5000) {
while(!kbhit()) { // MIENTRAS NO SE ELIJA OPCION
if(colfun==3)
ctrlservo(colfun,renfun,ZoomMax); // MANTIENE POSICION DE LA
// CAMARA MOVIL
almacena(spfile); // ALMACENA DATOS EN EL ARCHIVO
displaydatos(MODO); // DESPLIEGA DATOS EN PANTALLA
key.i=bioskey(0);
}
// SE DIGITARON TECLAS DE FLECHAS PARA MENU DE FUNCIONES
// O VENTANA DE IMAGEN DE CAMARA MOVIL
if((key.ch[0]!='\r'&&(key.ch[0]!=ESC)) {
// ELIMINA DESTELLO OPCION DE MENU O COORDENADA DE VENTANA ANTERIOR
if(colfun==3) blinking(0,colfun,renfun,10,sopcion); // UNBLINKING OPCION
else
blinking(0,colfun,renfun,1,sopcion); // UNBLINKING OPCION
}
// MUEVE CURSOR EN EL MENU O EN LA VENTANA DE IMAGEN
switch(key.i) {
case 0x4800: // UP
if(colfun== 3) { if(renfun== 7) renfun= 8; else renfun--; }
else { if(renfun== 6) renfun=12; else renfun--; }
break;
case 0x4b00: // LEFT
if(colfun== 3) { if(renfun== 7) renfun= 8; else renfun--; }
else { if(colfun==18) colfun=38; else colfun-=2; }
break;
case 0x4d00: // RIGHT
if(colfun== 3) { if(renfun== 8) renfun= 7; else renfun++; }
else { if(colfun==38) colfun=18; else colfun+=2; }
break;
case 0x5000: // DOWN
if(colfun== 3) { if(renfun== 8) renfun= 7; else renfun++; }
else { if(renfun==12) renfun= 6; else renfun++; }
break;
default: // TAB
if(colfun== 3) { colfun=28; renfun= 9; }
else { colfun= 3; renfun= 7; }
break; }
// DESTELLEA NUEVA OPCION DE MENU O COORDENADA DE VENTANA
if(colfun== 3) {
print( 3,10," tab ->",WHITE,BLUE);
blinking(1,colfun,renfun,10,sopcion); // BLINKING OPCION
scroll( 3,11,15,11,BLUE); // DESPLIEGA AYUDA EN LINEA
print( 3,11,onlinerec[opcion-0x30],LIGHTCYAN,BLUE); }
else {
print( 3,10," <- tab",WHITE,BLUE);
scroll( 3,11,15,11,BLUE); // DESPLIEGA AYUDA EN LINEA
print( 3,11,"Rastrea Zona",LIGHTCYAN,BLUE);
blinking(1,colfun,renfun,1,sopcion); // BLINKING OPCION
}
// SE ELIGIO ENTER
if(key.ch[0]!='\r') { // EJECUTA FUNCION SELECCIONADA
switch(opcion) {
case '1': ctrlservo(Rec); break; // GRABA EN CAMARAS
case '0': MODO=CAM; break; // SALE DE MODO REC
default: ctrlservo(colfun,renfun,ZoomMax); break; }
}
}

```

```

// SE ELIGIO OPCION DE ESCAPE
if(key.ch[0]==ESC) { MODO=CAM; key.i=0; }
} while(MODO==REC);
return;
// TERMINA CAMREC
/* -----
TERMINAN FUNCIONES DE LA CAMPAÑA DE DETECCIÓN DE IMAGENES
----- */
/* -----
FUNCIONES DE CONTROL DE RECEPTOR GPS
----- */
/* ENVIA LOS COMANDOS NECESARIOS PARA INICIALIZAR EL RECEPTOR GPS Y
LAS ORDENES REQUERIDAS PARA RECIBIR LA INFORMACION UTIL AL SISTEMA */
void startgps()
{
char *msg; // CADENA PUENTE DE MENSAJE
char *inf;

msg=malloc(100); *msg=0x00; inf=malloc(12); *inf=0x00;

comenta("Iniciando GPS ... ",1);
strcpy(msg,"$PMTL,00,1*"); txmsg(msg); delay(500); // ENVIA MASTER RESET

comenta("Datos por Omisión IS/N: ",1);
window( 2,15,39,18); gotoxy(col,ren); // SALTA A VENTANA DE MENSAJES
key.i=0;
while( key.ch[0]!='s'&&key.ch[0]!='S'&&
key.ch[0]!='n'&&key.ch[0]!='N'&&
key.ch[0]!=ESC ) key.i=bioskey(0); // SOLICITA RESPUESTA
window( 1, 1,80,25); // VUELVE A PANTALLA COMPLETA

if(key.ch[0]!='s'&&key.ch[0]!='S') { // DATOS DEL USUARIO

// SOLICITA LATITUD Y LONGITUD DEL SISTEMA
strcpy(msg,"$GPGGA,000000,"); // INICIA COMANDO
comenta("Latitud [1919.65,N]: ",0); leedato(inf, 7,"alnum");
strcpy(msg,inf); // AÑADE INFORMACION AL COMANDO
comenta("Longitud [09910.84,W]: ",0); leedato(inf,10,"alnum");
strcpy(msg,inf); // AÑADE INFORMACION AL COMANDO
strcpy(msg,"0,0,000,000,M,0000,M*"); // TERMINA COMANDO
txmsg(msg); // ENVIA LATITUD Y LONGITUD

// SOLICITA ALTITUD DEL SISTEMA, RESPECTO AL NIVEL DEL MAR
strcpy(msg,"$PMGLB,02,"); // INICIA COMANDO
comenta("Altitud [002244]: ",0); leedato(inf,6,"digit");
strcpy(msg,inf); strcpy(msg,"M*"); // TERMINA COMANDO
txmsg(msg); // ENVIA ALTITUD

// SOLICITA MAPA DATUM DEL SISTEMA, REVISAR MANUAL DEL RECEPTOR
strcpy(msg,"$PMTL,01,2,"); // INICIA COMANDO
comenta("Datum [47]: ",0); leedato(inf,2,"digit");
strcpy(msg,inf); strcpy(msg,"1,1,1*"); // TERMINA COMANDO
txmsg(msg); // ENVIA DATUM
}
else { // ENVIA DATOS POR OMISION
strcpy(msg,"$GPGGA,000000,1919.65,N,09910.84,W,0,0,000,000,M,0000,M*");
txmsg(msg); // LATITUD Y LONGITUD
strcpy(msg,"$PMGLB,02,002244,M*"); txmsg(msg); // ALTITUD
}
}

```

```

strncpy(msg,"$PGLS,01,2,47,1,1,1*"); txmsg(msg); // DATUM
comenta(" Latitud Longitud Altitud",1);
comenta("1919.65,W 09910.88,W 002244,M",1);
}
strncpy(msg,"$PGLD,00,3*"); txmsg(msg); // ESTABLECE MODO DE OPERACION

// SOLICITA INFORMACION UTIL AL SISTEMA AL RECEPTOR GPS
// PARA RECIBIRSE CADA SEGUNDO
strncpy(msg,"$PGL1,00,A00,2,A,*"); txmsg(msg); // A00: DIA Y HORA
strncpy(msg,"$PGL1,00,B00,2,A,*"); txmsg(msg); // B00: LATITUD Y LONGITUD
strncpy(msg,"$PGL1,00,B02,2,A,*"); txmsg(msg); // B02: ALTITUD EXPANDIDA
strncpy(msg,"$PGL1,00,E00,2,A,*"); txmsg(msg); // E00: CURSO Y VELOCIDAD
strncpy(msg,"$PGL1,00,F02,2,A,*"); txmsg(msg); // F02: SATELITES UTILIZADOS
strncpy(msg,"$PGL1,00,G00,2,A,*"); txmsg(msg); // G00: ERROR ESTIMADO
strncpy(msg,"$PGL1,00,H00,2,A,*"); txmsg(msg); // H00: ESTADO DEL GPS

comenta("GPS inicializado.",1);
free(msg);
return ;
} // TERMINA STARTGPS

/* RECIBE COMANDO ESPECIFICADO POR EL USUARIO PARA ENVIAR AL RECEPTOR GPS */
void comandgps()
{
char *mensaje; mensaje=malloc(100); *mensaje=0x00; // PARA RECIBIR COMANDO

// SOLICITA COMANDO AL USUARIO
comenta("Comando: ",0); leedata(mensaje,100,"gps");

if(*mensaje==0x00) return; // REvisa QUE HAYA COMANDO PARA EL RECEPTOR GPS

txmsg(mensaje); // TRANSMITE COMANDO DEL USUARIO AL RECEPTOR GPS
comenta("Comando transmitido.",1);

free(mensaje);
return ;
} // TERMINA COMANDGPS

/* ENVIA COMANDO PARA LIMPIAR LA MEMORIA DEL RECEPTOR GPS Y DATOS DEL
REGISTRO CORRESPONDIENTES AL GPS */
void cleargps(void)
{
// LIMPIA DATOS DE GPS DEL REGISTRO DE DATOS
reg.hora=0; reg.latitud=0; reg.error=0;
reg.longitud=0; reg.altitud=0;

// ENVIA COMANDO PARA LIMPIAR MEMORIA DEL RECEPTOR GPS
txmsg("$PGLM,00,1*"); delay(500); // ENVIA MASTER RESET

comenta("Memoria del GPS borrada.",1);
return ;
} // TERMINA CLEARGPS

/* CONTROLA MENU DE FUNCIONES DE CONTROL DEL RECEPTOR GPS */
void camgps(void)
{
// AYUDA EN LINEA DE LAS FUNCIONES DEL MENU DE CONTROL
char *onlinegps[4] = { "Sale de modulo GPS", "Habilita receptor GPS",
"Envia Comandos al GPS", "Borra Memoria del GPS" };

MODO=GPS;

// DIBUJA MENU DE FUNCIONES

```

```

scroll( 2, 7,24,11,BLUE); // LIMPIA VENTANA DE FUNCIONES
print( 3, 7,"1. ACTIVA RECEPTOR", LIGHTGREEN,BLUE);
print( 3, 8,"2. TRANSMITE COMANDOS", LIGHTGREEN,BLUE);
print( 3, 9,"3. LIMPIA MEMORIA", LIGHTGREEN,BLUE);
print( 3,10,"0. SALIR", LIGHTGREEN,BLUE);

// DESTELLEA PRIMERA OPCION DEL MENU DE FUNCIONES
colfun=3; renfun=7; blinking(1,colfun,renfun,20,6opcion); // BLINKING OPCION
print( 3,11,onlinegps[1],LIGHTCYAN,BLUE);

// MANTEN MENU DE FUNCIONES HASTA ELEGIR SALIR O ESCAPE
do {
key.i=0;
while(key.ch[0]!=ESC&&key.ch[0]!='\r'&&key.i!=0x4800&&
key.i!=0x4b00&&key.i!=0x4d00&&key.i!=0x5000) {
while(!kbhit()) { // MIENTRAS NO SE PIDE OPCION
displaydatos(MODO); // DESPLIEGA DATOS EN PANTALLA
key.i=bioskey(0);
}
// SI SE DIGITARON TECLAS DE FLECHAS
if((key.ch[0]!='\r')&&(key.ch[0]!=ESC)) {

// ELIMINA DESTELLEO DE OPCION ANTERIOR
blinking(0,colfun,renfun,20,6opcion); // UNBLINKING OPCION

// MUEVE EL CURSOR POR EL MENU DE FUNCIONES
switch(key.i) {
case 0x4800: if(renfun== 7) renfun=10; else renfun--; break; // UP
case 0x4b00: if(renfun== 7) renfun=10; else renfun--; break; // LEFT
case 0x4d00: if(renfun==10) renfun= 7; else renfun++; break; // RIGHT
case 0x5000: if(renfun==10) renfun= 7; else renfun++; break; // DOWN
default:: }

// DESTELLEA NUEVA OPCION Y DESPLIEGA AYUDA EN LINEA
blinking(1,colfun,renfun,20,6opcion); // BLINKING OPCION
scroll( 2,11,24,11,BLUE);
print( 3,11,onlinegps[opcion-0x30],LIGHTCYAN,BLUE);
}
// SE DIGITO ENTER, EJECUTA FUNCION CORRESPONDIENTE
if(key.ch[0]=='\r') {
switch(opcion) {
case '1': startgps(); delay(1500); break; // INICIALIZA GPS
case '2': comandgps(); delay(1500); break; // COMANDOS AL GPS
case '3': cleargps(); delay(1500); break; // LIMPIA MEMORIA DE GPS
case '0': MODO=CAM; break; // SALE DE MODO GPS
default:: }
}
// SE ELIGIO ESCAPE
if(key.ch[0]==ESC) { key.i=0; MODO=CAM; }
} while(MODO==GPS);

return;
} // TERMINA GPS

/*
=====
TERMINAN FUNCIONES DE CONTROL DE RECEPTOR GPS
=====*/

/*
=====
FUNCIONES QUE REDEFINEN LOS SERVICIOS DE INTERRUPCION IRQ6 E IRQ7
=====*/

```

```

/* REDEFINE SERVICIO DE INTERRUPCION IRQ7 (PUERTO PARALELO LPT1) PARA
RECIBIR PULSO DE SINCRONIA VERTICAL DE LAS IMAGENES, ESTABLECIENDO DOS
PULSOS POR IMAGEN */
void interrupt sincrnoimg()
{
  disable(); // BLOQUEA INTERRUPCIONES
  outportb(0x20, 0x67); // ANULA IRQ7 : PUERTO PARALELO 1

  // PROCESA NUEVO SERVICIO DE INTERRUPCION
  if(pulso) { // ES SEGUNDO PULSO DE SINCRONIA
    if(++byte1==0x80) { // ASEGURA RANGO DE BYTE 1 ( 00 -> 7F )
      byte1=0x00;
      if(++byte2==0x80) { // ASEGURA RANGO DE BYTE 2 ( 00 -> 7F )
        byte2=0x00;
        byte3++; // BYTE 3 DE 00 -> FF
      }
    }
    // CONSTRUYE NUMERO DE IMAGEN
    nocuadro=(byte3*0x1000)+(byte2*0x80)+byte1; // ( 00 00 00 -> FF FF FF )

    pulso=0x00; // ESPERA PRIMER PULSO DE SINCRONIA
    nobyteimg=0x00; // NUMERO DE BYTE DEL NUMERO DE IMAGEN

    // ACTIVA BANDERA DE DESPERTAR CAMARAS
    if(nocuadro>=(despierta+5000)) {
      outportb(Ncamera,0x0f); despierta=nocuadro;
    }
    outportb(0x21,0x20); // HABILITA IRQ4 : PUERTO SERIE 1
  } else pulso=0x01; // ESPERA SEGUNDO PULSO DE SINCRONIA

  enable(); // PERMITE INTERRUPCIONES
} // TERMINA VERTSYNC

/* REDEFINE SERVICIO DE INTERRUPCION IRQ4 (PUERTO SERIE COM1) PARA
TRANSMITIR INFORMACION A CAMA IMAGEN */
void interrupt txnoimagen()
{
  disable(); // BLOQUEA INTERRUPCIONES
  outportb(0x20, 0x68); // ANULA IRQ4 : PUERTO SERIE 1

  // TRANSMITE NO. DE IMAGEN
  switch(nobyteimg) {
    case 0: // BYTE DE CONTROL
      outportb(0x3f8,0x80); nobyteimg++; break;
    case 1: // BYTE 1
      outportb(0x3f8,byte1); nobyteimg++; break;
    case 2: // BYTE 2
      outportb(0x3f8,byte2); nobyteimg++; break;
    case 3: // BYTE 3
      outportb(0x3f8,byte3); nobyteimg++; break;
    case 4: // BYTE 4
      outportb(0x3f8,byte4); nobyteimg++; break;
    case 5: // BYTE 5
      outportb(0x3f8,byte5); nobyteimg++; break;
    case 6: // BYTE 6
      outportb(0x3f8,byte6); nobyteimg++; break;
    case 7: // BYTE 7
      outportb(0x3f8,byte7); nobyteimg++; break;
    case 8: // BYTE 8
      outportb(0x3f8,byte8); nobyteimg=0x00;
      outportb(0x21,0x30); // HABILITA IRQ7
      break;
    default : ;
  }
}

```

```

}
enable(); // PERMITE INTERRUPCIONES
} // TERMINA TXNOIMAGEN

/* REDEFINE SERVICIO DE INTERRUPCION IRQ3 (PUERTO SERIE COM2) PARA RECIBIR
DATOS DEL RECEPTOR GPS */
void interrupt rxdatosgps()
{
  disable(); // BLOQUEA INTERRUPCIONES
  outportb(0x20,0x63); // ANULA IRQ3 : PUERTO SERIE 2

  // RECIBE BYTE POR BYTE EL MENSAJE UNO DEL GPS
  if(nomsggps) {
    msggps1[chrgps]=inportb(0x2f8); // AÑADE BYTE AL MENSAJES
    msggps1[chrgps+1]=0x00;
    if(msggps1[chrgps]==0x0A) { // SI ES BYTE DE FIN DE MENSAJE
      chrgps=0; nomsggps=0x00; // TERMINA MENSAJE
      newmsggps=0x01; // ESTABLECE NUEVO MENSAJE DE GPS RECIBIDO
      else chrgps++; // SIGUIENTE BYTE
    }
  }

  // RECIBE BYTE POR BYTE EL MENSAJE DOS DEL GPS
  else {
    msggps2[chrgps]=inportb(0x2f8); // AÑADE BYTE AL MENSAJE
    msggps2[chrgps+1]=0x00;
    if(msggps2[chrgps]==0x0A) { // SI ES BYTE DE FIN DE MENSAJE
      chrgps=0; nomsggps=0x01; // TERMINA MENSAJE
      newmsggps=0x01; // ESTABLECE NUEVO MENSAJE DE GPS RECIBIDO
      else chrgps++; // SIGUIENTE BYTE
    }
  }

  enable(); // PERMITE INTERRUPCIONES
} // TERMINA RXDATOSGPS

/* -----
TERMINAN FUNCIONES QUE REDEFINEN LOS SERVICIOS DE INTERRUPCION IRQ4, IRQ7
----- */

/* -----
FUNCIONES DE COMUNICACION CON EL RECEPTOR GPS
----- */

/* TRANSMITE MENSAJE AL RECEPTOR GPS, GENERANDO CHECKSUM DE VERIFICACION
MENSAJE, INFORMACION A TRANSMITIR AL RECEPTOR GPS */
void txmsg(char *mensaje)
{
  char *sumchk; // PARA EL CHECKSUM DEL MENSAJE
  char *final; // PARA BLOQUE DE FIN DE MENSAJE
  unsigned status; // PARA CONOCER EL ESTADO DEL PUERTO SERIE

  sumchk=malloc(3); final=malloc(3); *sumchk=0x00;
  *final=0x0D; *(final+1)=0x0A; *(final+2)=0x00; // FIN DE MENSAJE

  // GENERA CHECKSUM DE VERIFICACION Y AÑADELO AL MENSAJE
  generachksm(mensaje,sumchk); // GENERA CHECKSUM
  strcat(mensaje,sumchk); // AÑADE CHECKSUM
  strcat(mensaje,final); // AÑADE FIN DE MENSAJE

  // TRANSMITE INFORMACION BYTE POR BYTE
  while(*mensaje!=0x00) {
    status=inportb(0x2fd); // LEE ESTADO DEL PUERTO SERIE 2
    if(status&0x01) // VERIFICA QUE NO HAYA ERROR DE PARIDAD
      outportb(0x2f8,*mensaje++); // TRANSMITE BYTE DE INFORMACION
  }
}

```

```

}
delay(200); // ESPERA TIEMPO DE RESPUESTA DEL RECEPTOR

free(sumchk); free(final);
return;
} // TERMINA TRMSG

/* GENERA 2 BYTES DE CHECKSUM DE VERIFICACION PARA EL MENSAJE DEL RECEPTOR
GPS
MENSAJE, INFORMACION PARA EL RECEPTOR GPS
CHECKSUM, CODIGO DE VERIFICACION DEL MENSAJE PARA EL RECEPTOR GPS */
void generachksun(char mensaje[100], unsigned char chksun[3])
{
    int indice; // PARA CICLO DE CONTEO

    chksun[0]=mensaje[1];

    // GENERA BYTE BINARIO DE CHECKSUM REALIZANDO LA OPERACION XOR ENTRE TODOS
    // LOS BYTES QUE COMPONEN EL MENSAJE PARA EL GPS ENTRE LOS SIMBOLOS S Y *
    for(indice=2; mensaje[indice]!='*'; indice++)
        if(mensaje[indice]!='0x00') chksun[0] ^= mensaje[indice];

    // CONVIERTE CADA NIBBLE DEL BYTE DE CHECKSUM EN UN BYTE ASCII AISLADO

    // DIVIDE EN DOS EL BYTE DE CHECKSUM
    chksun[1]=chksun[0]; chksun[2]=0x00;
    chksun[0]=(chksun[0]>>4); chksun[1]=0x0f;

    // TRANSFORMA EL VALOR DE LOS BYTES A SU CODIGO ASCII
    if(chksun[0]<0x0A) chksun[0]+=0x30; else chksun[0]+=0x37;
    if(chksun[1]<0x0A) chksun[1]+=0x30; else chksun[1]+=0x37;

    return;
} // TERMINA GENERACHKSUN

/* REVISAR CODIGO DE VERIFICACION CHECKSUM DEL MENSAJE DEL GPS
SI EL CODIGO ES CORRECTO REGRESA UN UNO
SI EL CODIGO ES INCORRECTO REGRESA UN CERO
MENSAJE, INFORMACION A REVISAR */
int revisachksun(char mensaje[100])
{
    char chksun[3]; // PARA GENERAR NUEVO CHECKSUM
    char *apunta; // APUNTA A CADENA DE MENSAJE

    // CALCULA NUEVO CHECKSUM DE MENSAJE
    generachksun(mensaje, chksun);

    // UBICA LA POSICION DEL CHECKSUM ORIGINAL EN EL MENSAJE RECIBIDO
    apunta=strchr(mensaje, '*');

    // COMPARA CODIGOS DE VERIFICACION
    if(apunta)
        if((chksun[0]==*(apunta+1)) && (chksun[1]==*(apunta+2))) return(1);

    return(0);
} // TERMINA REVISACHKSUN

/* -----
TERMINAN FUNCIONES DE COMUNICACION CON EL RECEPTOR GPS
----- */

/* MANEJA LAS FUNCIONES DEL MENU DE CONTROL DEL MODO DE OPERACION EN

```

```

VUELO
void cam(void)
{
    // AYUDA EN LINEA DE LAS OPCIONES DEL MENU DE FUNCIONES
    char *onlinecam[3] = {"Sale de Modo de Vuelo",
                        "Grabación de Imagenes", "Manipula el Receptor "};

    void interrupt txnoimagen(); // NUEVOS SERVICIOS DE INTERRUPTACION
    void interrupt sincrnoimg();
    void interrupt rxdatosgps();

    void interrupt (*oldcom1)(); // APUNTADORES A SERVICIOS DE INTERRUPTACION
    void interrupt (*oldlpt1)();
    void interrupt (*oldcom2)();

    startvar{CAM}; // INICIALIZA VARIABLES

    outportb(Control,0x00); // LIMPIA REG. PARA CONTROL RENOTO
    outportb(Wcamara,0x00); // LIMPIA REG. PARA NO. DE CAMARA

    MODO=CAM;

    scroll( 2,15,39,18,BLUE); // LIMPIA VENTANA DE COMENTARIOS

    abrefile(file,apfile); // ABRE ARCHIVOS

    // REVISAR QUE HAYA ARCHIVO PARA DATOS GENERADOS POR EL SISTEMA
    if(file[0]!='0x00') {
        comenta("SE REQUIERA ARCHIVO PARA DATOS",1);
        sound(1000); delay(500); nosound(); delay(3000); return; }

    // INICIALIZA LAS CAMARAS EN STANDBY, MODO DE ESPERA
    for(indice=0; indice<CAMARAS-1; indice++) {
        fcam[indice].code=STBY; fcam[indice].name=" STBY ";
        gotoxy(42+(indice*9),15); cprintf("%8s",fcam[indice].name); }

    oldcom1=getvect(0x0c); // GUARDA SERVICIOS DE INTERRUPTACION ORIGINALES
    oldcom2=getvect(0x0b);
    oldlpt1=getvect(0x0f);

    setvect(0x0c,txnoimagen); // ESTABLECE NUEVOS SERVICIOS DE INTERRUPTACION
    setvect(0x0b,rxdatosgps);
    setvect(0x0f,sincrnoimg);

    outportb(0x21,0x80); // HABILITA IRQ3: PUERTO SERIE 2

    puertoserie(1,2400,2,2,3); // COM 1 PARA TRANSMITIR NO. DE IMAGEN
    puertoserie(2,9600,1,2,2); // COM 2 PARA RECIBIR INFORMACION DEL GPS

    startgps(); // INICIALIZA GPS

    // DESPLIEGA MENU DE FUNCIONES
    scroll(-2, 7,24,11,BLUE);
    print( 3, 7,"1.CAMARAS", LIGHTGREEN,BLUE);
    print( 3, 8,"2.RECEPTOR GPS", LIGHTGREEN,BLUE);
    print( 3, 9,"0.SALIR", LIGHTGREEN,BLUE);

    // DESTELLA PRIMERA OPCION DEL MENU Y DESPLIEGA SU AYUDA EN LINEA
    colfun=3; renfun=7; blinking(1,colfun,renfun,13,opcion); // BLINKING OPCION
    print( 3,11,onlinecam[1],LIGHTCYAN,BLUE);

    // MANTIENE MENU DE FUNCIONES HASTA NO ELEGIR SALIR O ESCAPE
    do {

```

```

key.i=0;
while(key.ch[0]!=ESC&&key.ch[0]!='\r'&&key.i!=0x4800&&
key.i!=0x4b00&&key.i!=0x4d00&&key.i!=0x5000) {
// MIENTRAS NO SE ELIGA ALGUNA OPCION
while(!kbhit()) {
displaydatos(MODO); } // DESPLIEGA DATOS EN DISCO
key.i=bioskey(0);
}
// SE DIGITARON TECLAS DE FLECHA
if((key.ch[0]!='\r')&&(key.ch[0]!=ESC)) {

ELIMINA DESTELLO DE OPCION ANTERIOR
blinking(0,colfun,renfun,13,sopcion); // UNBLINKING OPCION

// MUEVE EL CURSOR POR EL MENU DE FUNCIONES
switch(key.i) {
case 0x4800: if(renfun== 7) renfun= 9; else renfun--; break; // UP
case 0x4b00: if(renfun== 7) renfun= 9; else renfun--; break; // LEFT
case 0x4d00: if(renfun== 9) renfun= 7; else renfun++; break; // RIGHT
case 0x5000: if(renfun== 9) renfun= 7; else renfun++; break; // DOWN
default:; }

// DESTELLEA NUEVA OPCION Y DESPLIEGA SU AYUDA EN LINEA
blinking(1,colfun,renfun,13,sopcion); // BLINKING OPCION
scroll( 3,11,24, 11,BLUE);
print( 3,11,onlinecam[sopcion-0x30],L(GHTCYAN,BLUE));
}
// SE DIGITO ENTER
if(key.ch[0]=='\r') {
// EJECUTA LA OPCION SELECCIONADA
switch(sopcion) {
case '1': camrec(); break; // FUNCIONES DE CAMARAS
case '2': camgps[1]; break; // FUNCIONES DE GPS
case '0': MODO=OFF; break; // SALE DE MODO CAM
default:; }

// DESPLIEGA MENU DE FUNCIONES AL REGRESAR
marco( 1, 6,25,12,WHITE,BLUE); print( 3, 6,"Controles",YELLOW,BLUE);
scroll( 2, 7,24,11,BLUE); // LIMPIA VENTANA DE FUNCIONES
print( 3, 7,"1.CAMARAS",LIGHTGREEN,BLUE);
print( 3, 8,"2.RECEPTOR GPS",LIGHTGREEN,BLUE);
print( 3, 9,"0.SALIR",LIGHTGREEN,BLUE);

// DESTELLEA PRIMERA OPCION DEL MENU Y DESPLIEGA SU AYUDA EN LINEA
colfun=3; renfun=7;
blinking(1,colfun,renfun,13,sopcion); // BLINKING OPCION
print( 3,11,onlinecam[1],LIGHTCYAN,BLUE);
}
if(key.ch[0]==ESC) MODO=OFF;

} while(MODO==CAM);

outportb(0x21,0x88); // RESTAURA CONTROLADOR DE INTERRUPCIONES

setvect(0x0c,oldcom); // RESTAURA SERVICIOS DE INTERRUPCIONES ORIGINALES
setvect(0x0b,oldcom2);
setvect(0x0f,oldipt1);

closefile(file,spfile); // CIERRA ARCHIVO
puertoserie(1,9600,1,1,1); // RESTAURA PUERTOS SERIE
puertoserie(2,9600,1,1,1);

free(file); free(vgps.hora); free(vgps.fecha);

return;
} // TERMINA CAM

/*-----
TERMINAN FUNCIONES EXCLUSIVAS DEL MODO DE OPERACION EN VUELO
*/

/* PROGRAMA PRINCIPAL DEL SISTEMA */
void main(void)
{
// AYUDA EN LINEA DE LAS OPCIONES DEL MENU DE FUNCIONES
char *onlinemain[3]={"Termina Programa",
"Operación en Vuelo", "Operación en Tierra"};
textattr((BLUE<<4)|LIGHTGREEN); // ATRIBUTOS PARA CPRINTF Y CSCAMF

// SOLICITA NUMERO DE CAMARAS ACTIVAS EN EL SISTEMA
do {
clrscr(); sound(1000); delay(100); nosound();
gotoxy(25,11); cprintf("Número de camaras en el sistema: ");
cscanf("%d",&CAMARAS);
} while(CAMARAS<1 || CAMARAS>4);
// SOLICITA NUMERO DE SENSORES HABILITADOS EN EL SISTEMA
do {
sound(1000); delay(100); nosound();
gotoxy(25,13); cprintf("Número de sensores en el sistema: ");
cscanf("%d",&SENSORES);
} while(SENSORES<1 || SENSORES>16);
startdisplay(); // DESPLIEGA PANTALLA DE INICIO

// DESTELLEA PRIMERA OPCION DEL MENU Y DESPLIEGA SU AYUDA EN LINEA
colfun=59; renfun=5;
blinking(1,colfun,renfun,15,sopcion); // BLINKING OPCION
scroll(56,11,75,11,LIGHTGRAY); print(57,11,onlinemain[1],BLACK,LIGHTGRAY);

// MANTIENE MENU HASTA QUE NO SE ELIJA SALIR O ESCAPE
do {
key.i=0;
while(key.ch[0]!=ESC&&key.ch[0]!='\r'&&
key.i!=0x4800&&key.i!=0x5000) key.i=bioskey(0);

// SE DIGITARON TECLAS DE FLECHA
if((key.ch[0]!='\r')&&(key.ch[0]!=ESC)) {

// ELIMINA DESTELLO DE OPCION ANTERIOR
blinking(0,colfun,renfun,15,sopcion); // UNBLINKING OPCION

// MUEVE EL CURSOR POR EL MENU DE FUNCIONES
switch(key.i) {
case 0x4800: if(renfun== 5) renfun= 7; else renfun--; break; // UP
case 0x5000: if(renfun== 7) renfun= 5; else renfun++; break; // DOWN
default:; }

// DESTELLEA NUEVA OPCION Y DESPLIEGA SU AYUDA EN LINEA
blinking(1,colfun,renfun,15,sopcion); // BLINKING OPCION
scroll(56,11,75,11,LIGHTGRAY); // LIMPIA AYUDA EN LINEA
print(57,11,onlinemain[sopcion-0x30],BLACK,LIGHTGRAY); // AYUDA EN LINEA
}
// SE DIGITO ENTER
if(key.ch[0]=='\r') {
// EJECUTA LA OPCION SELECCIONADA
switch(sopcion) {
case '1': worddisplay("cam"); cam(); key.i=0; break; // MODO VUELO

```

```

key.i=0;
while(key.ch[0]!=ESC&&key.ch[0]!='\r'&&key.i!=0x4800&&
key.i!=0x4b00&&key.i!=0x4d00&&key.i!=0x5000) {
// MIENTRAS NO SE ELIGA ALGUNA OPCION
while(!kbhit()) {
displaydatos(MODO); } // DESPLIEGA DATOS EN DISCO
key.i=bioskey(0);
}
// SE DIGITARON TECLAS DE FLECHA
if((key.ch[0]!='\r')&&(key.ch[0]!=ESC)) {

ELIMINA DESTELLO DE OPCION ANTERIOR
blinking(0,colfun,renfun,13,opcion); // UNBLINKING OPCION

// MUEVE EL CURSOR POR EL MENU DE FUNCIONES
switch(key.i) {
case 0x4800: if(renfun== 7) renfun= 9; else renfun--; break; // UP
case 0x4b00: if(renfun== 7) renfun= 9; else renfun--; break; // LEFT
case 0x4d00: if(renfun== 9) renfun= 7; else renfun++; break; // RIGHT
case 0x5000: if(renfun== 9) renfun= 7; else renfun++; break; // DOWN
default:: }

// DESTELLEA NUEVA OPCION Y DESPLIEGA SU AYUDA EN LINEA
blinking(1,colfun,renfun,13,opcion); // BLINKING OPCION
scroll( 3,11,26,11,BLUE);
print( 3,11,onlinecam|opcion-0x10|,LIGHTCYAN,BLUE);
}
// SE DIGITO ENTER
if(key.ch[0]=='\r') {
// EJECUTA LA OPCION SELECCIONADA
switch(opcion) {
case '1': camrec(); break; // FUNCIONES DE CAMARAS
case '2': camgps(); break; // FUNCIONES DE GPS
case '0': MODO=OFF; break; // SALE DE MODO CAM
default:: }

// DESPLIEGA MENU DE FUNCIONES AL REGRESAR
marco( 1, 6,25,12,WHITE,BLUE); print( 3, 6,"Controles",YELLOW,BLUE);
scroll( 2, 7,24,11,BLUE); // LIMPIA VENTANA DE FUNCIONES
print( 3, 7,"1.CAMARAS", LIGHTGREEN,BLUE);
print( 3, 8,"2.RECEPTOR GPS", LIGHTGREEN,BLUE);
print( 3, 9,"0.SALIR", LIGHTGREEN,BLUE);

// DESTELLEA PRIMERA OPCION DEL MENU Y DESPLIEGA SU AYUDA EN LINEA
colfun=3; renfun=7;
blinking(1,colfun,renfun,13,opcion); // BLINKING OPCION
print( 3,11,onlinecam[1],LIGHTCYAN,BLUE);
}
if(key.ch[0]==ESC) MODO=OFF;
}
while(MODO==CAM):
outportb(0x21,0x88); // RESTAURA CONTROLADOR DE INTERRUPCIONES

setvect(0x0c,oldcom1); // RESTAURA SERVICIOS DE INTERRUPCION ORIGINALES
setvect(0x0b,oldcom2);
setvect(0x0f,oldipt1);

closefile(file,spfile); // CIERRA ARCHIVO
puertoserie(1,9600,1,1,1); // RESTAURA PUERTOS SERIE
puertoserie(2,9600,1,1,1);

free(file); free(vgps.hora); free(vgps.fecha);

```

```

return;
} // TERMINA CAM

/*-----
* TERMINAR FUNCIONES EXCLUSIVAS DEL MODO DE OPERACION EN VUELO
*-----*/

/* PROGRAMA PRINCIPAL DEL SISTEMA */
void main(void)
{
// AYUDA EN LINEA DE LAS OPCIONES DEL MENU DE FUNCIONES
char *onlinemain[3]={ "Termina Programa ",
"Operación en Vuelo ", "Operación en Tierra" };
textattr((BLUE<<4)|LIGHTGREEN); // ATRIBUTOS PARA CPRINTF Y CSCAMF

// SOLICITA NUMERO DE CAMARAS ACTIVAS EN EL SISTEMA
do {
clrscr(); sound[1000]; delay(100); nosound();
gotoxy(25,11); cprintf(" Número de camaras en el sistema: ");
cscanf("%d",&CAMARAS);
{ while(CAMARAS<1 || CAMARAS>4):
// SOLICITA NUMERO DE SENSORES HABILITADOS EN EL SISTEMA
do {
sound[1000]; delay(100); nosound();
gotoxy(25,13); cprintf("
");
gotoxy(25,13); cprintf("Número de sensores en el sistema: ");
cscanf("%d",&SENSORES);
} while(SENSORES<1 || SENSORES>16);
startdisplay(); // DESPLIEGA PANTALLA DE INICIO

// DESTELLEA PRIMERA OPCION DEL MENU Y DESPLIEGA SU AYUDA EN LINEA
colfun=5; renfun=5;
blinking(1,colfun,renfun,15,opcion); // BLINKING OPCION
scroll(56,11,75,11,LIGHTGRAY); print(57,11,onlinemain[1],BLACK,LIGHTGRAY);

// MANTIENE MENU HASTA QUE NO SE ELIJA SALIR O ESCAPE
do {
key.i=0;
while(key.ch[0]!=ESC&&key.ch[0]!='\r'&&
key.i!=0x4800&&key.i!=0x5000) key.i=bioskey(0);

// SE DIGITARON TECLAS DE FLECHA
if((key.ch[0]!='\r')&&(key.ch[0]!=ESC)) {

// ELIMINA DESTELLO DE OPCION ANTERIOR
blinking(0,colfun,renfun,15,opcion); // UNBLINKING OPCION

// MUEVE EL CURSOR POR EL MENU DE FUNCIONES
switch(key.i) {
case 0x4800: if(renfun== 5) renfun= 7; else renfun--; break; // UP
case 0x5000: if(renfun== 7) renfun= 5; else renfun++; break; // DOWN
default:: }

// DESTELLEA NUEVA OPCION Y DESPLIEGA SU AYUDA EN LINEA
blinking(1,colfun,renfun,15,opcion); // BLINKING OPCION
scroll(56,11,75,11,LIGHTGRAY); // LIMPIA AYUDA EN LINEA
print(57,11,onlinemain|opcion-0x10|,BLACK,LIGHTGRAY); // AYUDA EN LINEA
}
// SE DIGITO ENTER
if(key.ch[0]=='\r') {
// EJECUTA LA OPCION SELECCIONADA
switch(opcion) {
case '1': workdisplay("cam"); cam(); key.i=0; break; // MODO VUELO

```

```

case '2': workdisplay("vtr"); vtr(); key.i=0; break; // MODO TIERRA
case '0': key.ch[0]=ESC; break; // TERMINA PROGRAMA
default:; }

// AL REGRESAR RESTAURA PANTALLA INICIAL
// DESTELLEA PRIMERA OPCION DEL MENU Y DESPLIEGA SU AYUDA EN LINEA
if(key.ch[0]!=ESC) {
key.i=0;
startdisplay(); // RESTAURA PANTALLA INICIAL
colfun=59; renfun=5;
blinking(1,colfun,renfun,15,opcion); // BLINKING OPCION
scroll(56,11,75,11,LIGHTGRAY);
print(57,11,onlinemain(1),BLACK,LIGHTGRAY); gotoxy(colfun,renfun); }
}
while(key.ch[0]!=ESC);

textattr((BLACK<<4)|LIGHTGRAY); // ATRIBUTOS PARA CPRINTF Y CSCAMF
clrscr();
} // TERMINA MAIN

```

```

case '2': workdisplay("vtr"); vtr(); key.i=0; break; // MODO TIERRA
case '0': key.ch[0]=ESC; break; // TERMINA PROGRAMA
default:: }

// AL REGRESAR RESTAURA PANTALLA INICIAL
// DESTELLA PRIMERA OPCION DEL MENU Y DESPLIEGA SU AYUDA EN LINEA
if(key.ch[0]!=ESC) {
key.i=0;
startdisplay(); // RESTAURA PANTALLA INICIAL
colfun=59; renfun=5;
blinking(1,colfun,renfun,15,6opcion); // BLINKING OPCION
scroll(56,11,75,11,LIGHTGRAY);
print(57,11,onlinemain[1],BLACK,LIGHTGRAY); gotoxy(colfun,renfun); }
} while(key.ch[0]!=ESC);

textattr((BLACK<<6)|LIGHTGRAY); // ATRIBUTOS PARA CPRINTF Y CSCANSF
clrscr();
} // TERMINA MAIN

```

REFERENCIAS

- [1] Peralta R. et. al. Sistema para obtención de imágenes aéreas multiespectrales de bajo costo. VI Congreso Nacional de Instrumentación. Guanajuato, Gto. Septiembre, 1990. Memorias del Congreso. Trabajo No. 20.
- [2] Peralta R. et. al. Grupos de cámaras CCD multiespectrales para diversos estudios de PR con bajo costo. V Simposio Latinoamericano de Percepción Remota SELPER. Cusco, Perú. Octubre 1991. Memorias tomo II. pp 633-643.
- [3] Peralta R. et. al. CCD Image acquisition for multiespectral teledetection SPIE Symposium on Electronic Imaging: Science and Technology. San José, California USA. Febrero 1992. SPIE proceeding Vol. 1656 pp 350-358.
- [4] Navarrete M., Segura F., Vaquera E., Peralta R., Contenedor aeroportado para percepción remota. IX Congreso Nacional de la Sociedad Mexicana de Instrumentación. Cancún, Quintana Roo. Septiembre, 1994. Memorias del Congreso, pag. 435-439.
- [5] Lewis C. Eggerbrecht, Interfacing to the IBM personal Computer, 2a. edición, 1994, SAMS, USA, 345 pp.
- [6] Navarrete M., Hernández M. A., Domínguez R., Contreras J., Prado J., Electrónica de control de un sistema aeroportado para obtención de imágenes de video. X Congreso Nacional de la Sociedad Mexicana de Instrumentación. Xalapa, Ver. Septiembre, 1995. Memorias del Congreso. Trabajo No. 26.

- [7] Wilfredo Martínez Payán, Control computacional de cámaras multiespectrales aerotransportadas para percepción remota. Tesis de Licenciatura. Facultad de Ingeniería. UNAM. 1992.**
- [8] Jacinto Elías Vaquera Rojas. Integración de sensores para automatización aeronáutica. Tesis de Licenciatura. Facultad de Ingeniería. UNAM. Marzo de 1994.**
- [9] National Semiconductor. Special Purpose Linear Device Databook, 1989.**
- [10] National Semiconductor. Data Acquisition Linear Device Databook. 1989.**
- [11] Analog Device. Data Conversion Products Databook. 1989.**
- [12] Peralta R. Aeronave de Control Remoto para Teledetección. Instituto de Ingeniería-PUIDE/UNAM. 1988.**