



**UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO**

**PROGRAMA DE MAESTRÍA Y DOCTORADO EN
INGENIERÍA**

**DISEÑO E IMPLEMENTACIÓN DE UN
VEHÍCULO AÉREO NO TRIPULADO**

QUE PARA OBTAR EL GRADO DE:

MAESTRO EN INGENIERÍA

INGENIERÍA ELÉCTRICA – SISTEMAS ELECTRÓNICOS

P R E S E N T A

ING. ARMANDO PÉREZ RAMÍREZ

DIRECTOR DE TESIS:

M.I. JESÚS ÁLVAREZ CASTILLO

ENERO 2012



Jurado Asignado

Presidente: Dra. Navarrete Montesinos Margarita

Secretario: Dr. Peña Cabrera Mario

Vocal: M. I. Álvarez Castillo Jesús

1er. Suplente: M.I. Haro Ruíz Luis Arturo

2do. Suplente: Dr. Prado Molina Jorge

DIRECTOR DE TESIS:

M. I. Álvarez Castillo Jesús

Agradecimientos

La presente Tesis es un esfuerzo en el cual, directa o indirectamente, participaron varias personas leyendo, opinando, corrigiendo, teniéndome paciencia, dando ánimo, acompañándome en los momentos de felicidad.

Quiero expresar mi agradecimiento

A mi Director de Tesis, M.I. Jesús Álvarez Castillo por brindarme la oportunidad de recurrir a su capacidad y experiencia. Por haber confiado en mí persona, por la paciencia y por la dirección de este trabajo.

A la Universidad Nacional Autónoma de México, UNAM, la Máxima Casa de Estudios, y a la Coordinación de Estudios de Posgrado que con el apoyo de una beca me permitieron desarrollar este trabajo de tesis y mis estudios de maestría, adquirí mucha experiencia profesional.

A mi mamá con el más sincero cariño porque gracias a su apoyo, consejos y fortaleza he llegado a realizar una de mis grandes metas lo cual constituye la herencia más valiosa que pudiera recibir.

A mi padre que aunque no esté ya entre nosotros sigue vivo en mi pensamiento; y que su estímulo fue mi impulso para llegar al final, que mientras vivió fue un apoyo incondicional en mis estudios y estuvo atento de los mismos, ambos han compartido mis logros profesionales y personales.

A mis hermanos por ser parte importante de mi vida y representar la unidad familiar.

Gracias también a mis queridos compañeros, que me apoyaron y me permitieron entrar en su vida durante este tiempo.

Y a todas aquellas personas que de una u otra forma, colaboraron o participaron en la realización de esta investigación, hago extensivo mi más sincero agradecimiento.

Para mí, la cosa más importante en este mundo no es dónde nos encontramos sino el rumbo al que nos dirigimos. Para llegar al puerto del paraíso a veces tenemos que navegar a favor del viento y a veces en contra, pero debemos navegar y no ir a la deriva y tampoco estar anclados.

Oliver Wendell Holmes, Jr.

Índice General

<i>AGRADECIMIENTOS</i>	4
<i>QUIERO EXPRESAR MI AGRADECIMIENTO</i>	4
ÍNDICE DE FIGURAS	9
ÍNDICE DE TABLAS	12
RESUMEN	13
I INTRODUCCIÓN, OBJETIVOS Y CONTENIDO	14
1.1 JUSTIFICACIÓN:	14
1.2 OBJETIVOS	15
1.3 CONTENIDO DE LA TESIS	16
II CONCEPTOS BÁSICOS	17
2.1 INTRODUCCIÓN	17
2.2 DEFINICIONES CLAVES:	17
2.17 FUNCIONAMIENTO Y ELECTRÓNICA BÁSICA EN AEROMODELOS DE R/C.....	17
III CONSTRUCCIÓN E IMPLEMENTACIÓN DEL PROTOTIPO	19
3.1 INTRODUCCIÓN:	19
3.3.1 <i>Instrumentos de Vuelo</i>	19
3.3.2 <i>Instrumentos de navegación</i>	20
3.4 DISEÑO E INTEGRACIÓN ELECTRÓNICA.....	22
3.5 SENSORES	22
3.5.1 <i>Sensor de Velocidad</i>	23
3.5.2 <i>Sensor de Altitud</i>	32
3.5.3 <i>Sensor de orientación</i>	35
3.5.4 <i>Sensor de humedad</i>	36
3.5.5 <i>Sensor de temperatura</i>	37
3.5.6 <i>GPS</i>	38
3.6 IMPLEMENTACIÓN DE LA COMPUTADORA DE VUELO O FMS (FLIGHT MANAGEMENT SYSTEM).	42
3.6.1 <i>Descripción de la computadora de vuelo</i>	43
3.6.2 <i>Tarjeta de telemetría</i>	44
3.6.3 <i>Definiendo reglas de comunicación</i>	47
3.6.4 <i>Piloto automático</i>	51
3.6.5 <i>Teoría para el control de servomotores</i>	51

3.6.6 Método para el control de 3 o más servomotores.....	52
3.6.7 Control automático de alerones y estabilizador horizontal.....	53
3.6.8 Control PID de alabeo y cabeceo.....	56
3.6.9 Sistema de seguridad anti fallos (Fail-Safe System).	59
3.7 ESTACIÓN DE CONTROL TERRENA.	61
3.7.1 Descripción del modulo de comunicaciones.	62
3.7.2 Especificaciones del software G.C.S.	64
3.7.3 Desarrollo del software.....	64
IV REALIZACIÓN DE PRUEBAS Y ADQUISICIÓN DE DATOS.	67
V RESULTADOS DE LAS PRUEBAS	71
VI CONCLUSIONES	74
6.1 TRABAJO FUTURO.....	75
BIBLIOGRAFÍA.	76
APÉNDICE I	79
I BREVE HISTORIA DE LA AVIACIÓN Y PERSPECTIVA DE LOS UAV.	80
1.1 Primeros diseños de UAV	80
1.2 Actualidades en los UAVs.....	81
II PRINCIPIOS BÁSICOS DE VUELO.	84
2.1 Sistema de Ejes.....	85
2.2 Fuerzas ejercidas sobre el avión.....	86
2.3 Equilibrio y compensación fina (trimming) de un avión.....	89
2.4 Estabilidad estática y dinámica.....	89
III SUPERFICIES DE CONTROL DE UNA AERONAVE Y ESTRUCTURA.....	91
IV ESTRUCTURA DEL AVIÓN.	92
V ETAPAS DE VUELO.....	93
VI ECUACIONES BÁSICAS DE MOVIMIENTO DE UN AEROPLANO.	94
VII CARACTERÍSTICAS DE LOS AVIONES EMPLEADOS: SR. FALCÓN Y HOBBYSTAR 60	98
6.1 Aviónica.....	100
6.2 Instrumentación aeronáutica.....	100
ANEXOS	101
ANEXO I	102
DIAGRAMAS ESQUEMÁTICOS(D.E), LISTA DE MATERIALES(L.M) Y MASCARÁS DE COMPONENTES(M.C.).....	102
D.E. ACONDICIONAMIENTO DEL TUBO DE PITOT.	102
Lista de Materiales.....	103

<i>Máscaras de Componentes</i>	103
<i>Ubicación de Componentes Vista Superior</i>	103
DIAGRAMA ESQUEMÁTICO TARJETA GPS.	104
<i>Lista de Materiales</i>	104
<i>Vista Superior Tarjeta GPS</i>	105
<i>Ubicación de Componentes</i>	105
DIAGRAMA ESQUEMÁTICO TARJETA DE TELEMETRÍA	106
<i>Lista de Materiales.</i>	107
<i>Vista Superior tarjeta de Telemetría.</i>	108
<i>Ubicación de Componentes</i>	108
DIAGRAMA ESQUEMÁTICO PILOTO AUTOMÁTICO	109
<i>Lista de Materiales.</i>	110
<i>Vista Superior Tarjeta de Piloto Automático</i>	110
<i>Ubicación de Componentes Piloto Automático</i>	111
DIAGRAMA ESQUEMÁTICO F.S.S.....	111
<i>Lista de Materiales</i>	112
<i>Vista Inferior F.S.S.</i>	112
<i>Ubicación de Componentes F.S.S.</i>	112
ANEXO II	113
<i>Rutinas de Programación Tarjeta de Telemetría</i>	113
<i>Rutinas de Programación Tarjeta de Piloto Automático.</i>	119
<i>Rutina de Programación Brújula Digital.</i>	124

Índice de figuras

Figura 1. Funcionamiento Básico de un Avión de RC.	18
Figura 2. Ejemplo de aerovía Acapulco México. Imagen tomada del sitio IVAOMX: http://ivaomx.wordpress.com/2010/06/02/aerovias-de-baja-altitud-en-el-espacio-aereo-mexicano/	20
Figura 3. Funcionamiento GPS. Imagen tomada del sitio Geocienciasipc: http://geocienciasipc.blogspot.com/2011/07/nuevas-tecnologias-sistema-de-posicion.html	21
Figura 4. Diagrama básico sistema de telemetría y control automático.	22
Figura 5. Tubo de Pitot: presión dinámica en B, y presión estática en A.	23
Figura 6. Relación teórica: Rapidez vs Presión diferencial en el tubo de pitot.	25
Figura 7. Tubo de Pitot-1: presión estática, entrada (A) y salida (Po). Presión dinámica, entrada (B) y salida (Ps).	26
Figura 8. Tubos de Pitot fabricados.	26
Figura 9. Diagrama de bloques del sensor de presión MPX2010.	27
Figura 10. Salida de tensión vs Presión diferencial - Hoja de datos MPX2010.	27
Figura 11. Circuito acondicionador para el Tubo de Pitot: Entrada de tensión de la presión estática ($V+PE$) y dinámica ($V-PD$), $Vout$ = salida de presión diferencial hacia el microcontrolador.	28
Figura 12. Banco de pruebas - Túnel de viento.	29
Figura 13. Resultados de caracterización -Tubo de Pitot 1.	29
Figura 14. Resultados de caracterización -Tubo de Pitot 2.	30
Figura 15. Resultados de caracterización -Tubo de Pitot 3.	30
Figura 16. Tension de salida vs Presion -Hoja de Datos MPXAZ6115A.	33
Figura 17. Diagrama de Bloques Convertidor A/D MCP3425.	34
Figura 18. Esquema de conexiones empleado para el MCP34125.	35
Figura 19. Brújula digital modelo HMC6352 y dimensiones.	35
Figura 20. Sensor de Humedad HMZ-433.	36
Figura 21. Salida de Tensión vs Humedad Relativa a 25°C].	36
Figura 22. Diagrama de bloques y vista superior del sensor LM35.	37
Figura 23. Interfaz de conexión recomendada MCP9800.	38
Figura 24. Dimensiones del modulo GPS [mm].	39
Figura 25. Tarjeta de interfaz fabricada para el modulo GPS.	39
Figura 26. Diagrama de flujo para la lectura del GPS.	42
Figura 27. Diagrama de bloques del sistema UAV propuesto.	43
Figura 28. Realización de pruebas: Prototipo tarjeta de telemetría y GPS.	45
Figura 29. Diagrama de bloques para el envío de información.	45

Figura 30. Sentencias de datos entregadas por el modulo GPS-RXM-SR.....	46
Figura 31. Se muestra el envío de la sentencia GPRMC y la verificación de checksum enmarcadas por líneas.....	46
Figura 32. Envío de valores de los sensores.	47
Figura 33. Cara superior tarjeta de telemetría.....	48
Figura 34. Cara inferior tarjeta de telemetría.	49
Figura 35. Diagrama de flujo sistema de telemetría.	50
Figura 36. Señal PWM para el control de un servomotor.....	52
Figura 37. Ventana de tiempo para el control de 4 servomotores.	53
Figura 39. Diagrama de conexiones del acelerómetro MMA7360.	54
Figura 38 . Funcionamiento básico del acelerómetro.....	54
Figura 40. Ángulos de Roll, Pitch y Yaw mediante el acelerómetro.....	56
Figura 41. Diagrama de bloques control PID.....	57
Figura 42. Diagrama de bloques básico para el funcionamiento del piloto automático.	58
Figura 43. Circuito impreso – piloto automático.....	58
Figura 44. Diagrama de flujo para la programación del piloto automático.	59
Figura 45. F.S.S: en JP1 se conecta el canal auxiliar del radiocontrol mientras que JP2 envía la señal para desactivar la tarjeta del piloto automático.....	60
Figura 46. Diagrama de bloques F.S.S.	60
Figura 47.Switch ON- X=1[V/div], Y=1.2[ms/div].	61
Figura 48.Switch OFF- X=1[V/div], Y=2.45[ms/div].	61
Figura 49. Modulo HM-TR915.	62
Figura 50. Diagrama de bloques estación de control terrena.....	63
Figura 51. Configuración e inicialización del puerto serial.....	64
Figura 52. Separación de datos.	65
Figura 53. Separación de datos GPS.	66
Figura 54. Interfaz gráfica para la estación de control terrena.....	66
Figura 55. Verificación del aeromodelo en tierra.....	67
Figura 56. Adquisición de datos: 1. Datos de los sensores, 2. Datos GPS, 3. Sentencia errónea.	68
Figura 57. Mapa creado con GPS-Visualizer para las coordenadas: Lat = 1914.3198, N y Lon = 09913.3678, W.....	69
Figura 58. Interfaz grafica para la estación de control terrena mostrando el trazado de una ruta dentro del circuito de ciudad universitaria.....	70
Figura 59. Visualización de datos en la interfaz gráfica.	71
Figura 60. Vista 1: Trazado de la ruta y puntos de geolocalización del aeromodelo.....	72
Figura 61.Vista 2: Trazado de la ruta y puntos de geolocalización del aeromodelo.....	72

Figura 62. Rango máximo de alcance registrado mediante el modulo de comunicaciones HM-TR915.....	73
Figura 63. Tornillo aéreo – Leonardo Da Vinci.	80
Figura 64.El Seagul construido por Elbitsystems, Israel.	81
Figura 65.El RQ-2B Pioneer, diseñado por Pioneer UAV Inc. Operado por la marina de los Estados Unidos de América.	81
Figura 66.UAV Seagul	82
Figura 67.El DP-5X de DragonFly	82
Figura 68. Vehículo aéreo de reconocimiento de objetivos con microcámara.	83
Figura 69.Vehículo aéreo comparado con la escala de un lápiz.	83
Figura 70. Avioneta Cessna 172.....	84
Figura 71.Giros de un avión en torno a sus ejes.	85
Figura 72. Fuerzas ejercidas sobre un avión.	87
Figura 73. Flujo de aire en un perfil alar.....	87
Figura 74. Angulo de ataque.....	88
Figura 75. Relación entre ángulo de ataque y velocidad.	88
Figura 76. Acciones y movimientos al efectuar las acciones de compensación.	89
Figura 77.Desequilibrio debido a la pérdida de un motor.	90
Figura 78. Superficies de control de un avión: A.- timón de la dirección, B.- estabilizador vertical, C.- estabilizador horizontal, D.- elevador, E.- alerones, F.- fuselaje, G.- tren de aterrizaje, H.- alas.....	91
Figura 79. Ángulos en las superficies de control.	92
Figura 80. Cambio del ángulo de ataque en el aterrizaje.....	93
Figura 81. Dimensiones de los Aviones Sr. Falcón y Hobbistar 60.	99

Índice de Tablas

Tabla 1 Tipos de instrumentos de vuelo.	19
Tabla 2. Comparación de las pruebas de caracterización para el Tubo de Pitot # 3: mediciones de rapidez y tensión.	31
Tabla 3. Valores y descripción del enunciado GPRMC.	41
Tabla 4. Descripción de los identificadores.	48
Tabla 5. Parámetros del acelerómetro MMA7360.....	54
Tabla 6. Desglose de la cadena numero 1 (ver Fig.56).	68
Tabla 7. Características Sr. Falcon y Hobisstar 60	99

Resumen

Los vehículos aéreos no tripulados, conocidos como UAV (Unmanned Aerial Vehicle) son aeronaves con capacidad de realizar un vuelo en forma autónoma, sin un piloto, son usados en diversas aplicaciones civiles y militares, esto se logra mediante algoritmos de control o sistemas electrónicos redundantes.

Se implementó un F.M.S. para un aeromodelo de radiocontrol obteniendo así un sistema de bajo costo para emular un UAV. El sistema transmite el estado de algunas variables (posición geográfica, altitud, presión, humedad, temperatura y velocidad) para monitorear su desempeño, por otro lado se creó un piloto automático capaz de estabilizarlo en el aire en un 30% sin la ayuda de un piloto en tierra, dejando a este únicamente el control de la altitud, viraje y aceleración de forma manual.

Se diseñaron, implementaron e integraron los instrumentos de vuelo necesarios para poder mejorar la maniobrabilidad de un aeromodelo de R/C y asemejarse a un UAV. El F.M.S incluye una tarjeta encargada de la telemetría y otra dedicada al control de las superficies del aeromodelo, en conjunto las dos tarjetas tienen la función de enviar y recibir el estado de los sensores así como la recepción de comandos. Mediante la sección del piloto automático se posicionan las superficies de control para estabilizar el avión, o bien tomar el control desde tierra en caso de falla (F.S.S).

Se diseñó y programó la interfaz grafica para la Estación de Control Terrena con la cual se logró monitorear mediante instrumentos virtuales el valor de los diversos sensores, adicionalmente es posible ver las coordenadas geográficas del aeromodelo en un indicador, trazar la trayectoria seguida por este, así como ubicarlo en un mapa satelital.

I Introducción, objetivos y contenido.

Vehículos aéreos no tripulados conocidos como U.A.V (Unmanned Aerial Vehicle) es el acrónimo para denominar a una aeronave que es capaz de realizar un vuelo en forma autónoma.

En la mayor parte de los casos los UAV son operados por radiocontrol para el aterrizaje y el despegue, guiados mediante un GPS hasta una altura de seguridad para empezar a volar de manera autónoma.

Estos vuelos autónomos se logran mediante algoritmos de control y sistemas electrónicos redundantes. Hoy en día estos son usados en diversas aplicaciones civiles y militares [1].

1.1 Justificación:

El proyecto surge ante la necesidad de poder realizar vuelos de forma autónoma, esto permitirá desarrollar diversas aplicaciones como lo son: monitoreo de regiones geográficas de difícil acceso, monitoreo de vialidades, vigilancia de zonas de conservación ecológica, evaluación de desastres naturales.

Se enfocará en otorgar un nivel de instrumentación electrónica considerable, tal que podamos controlar el vehículo remotamente al mismo tiempo que vigilamos algunos parámetros. El nivel de instrumentación electrónica que se pueda alcanzar dependerá de la infraestructura y recursos tecnológicos disponibles.

Los resultados obtenidos, como el software que se implementará o la instrumentación electrónica encargada de la telemetría pueden ser empleados como punto de partida en otras aéreas de la ingeniería como telecomunicaciones, evaluación de algoritmos de control e inteligencia artificial, entre otras.

1.2 Objetivos

Los 3 Objetivos principales de este proyecto son los siguientes:

- Diseño de los instrumentos¹ necesarios para poder emular un UAV.
- Diseño de un F.M.S capaz de adquirir variables del aeromodelo para mejorar su desempeño en comparación a un modelo básico de aeromodelismo, así como ser capaz de enviar y recibir datos o comandos para modificar alguna función o bien controlarlo desde tierra.
- Diseño de una estación de control terrena que permita verificar el estado y ubicación del aeromodelo así como monitorear su desempeño.

Con el fin de alcanzar los objetivos mencionados, se usan sensores de presión, rapidez, altitud, posición, modulo GPS, módulos de comunicación etc. Se implementara el hardware y firmware que permita incorporar estos dispositivos.

¹ Los instrumentos pueden ser de vuelo (altímetro, indicador de rumbo magnético, velocímetro etc) o de navegación (GPS).

1.3 Contenido de La Tesis

A continuación se presenta un panorama general de cada capítulo:

En el primero se presenta una breve introducción acerca de vehículos aéreos no tripulados, el planteamiento del problema y los objetivos de la presente tesis.

En el capítulo dos se presentan algunos conceptos relacionados con los vehículos aéreos no tripulados.

En el capítulo 3 desarrolla la implementación de los instrumentos de vuelo, acondicionamiento de señales, fabricación de circuitos impresos, así como la emulación de una estación de control terrena.

En el capítulo 4 mostramos los datos adquiridos mediante las tarjetas desarrolladas, así como la interpretación de los datos obtenidos.

Finalmente en el capítulo 5 se realiza un análisis de resultados generales del proyecto, y las conclusiones y recomendaciones en el 6.

II Conceptos básicos.

2.1 Introducción

Con el propósito de hacer más eficientes los vehículos aéreos el hombre se ha enfocado en perfeccionarlos, estos tienen un gran campo de aplicaciones, desde un medio de transporte masivo, armas de combate o con fines de investigación como lo son los transbordadores espaciales, extendiéndose lo que conocemos como aeronáutica a la investigación aeroespacial. Los UAV, han tenido un gran auge en aplicaciones civiles y militares.

Los aviones fueron introducidos durante la primera guerra mundial (1917) teniendo como principal desarrollador a la industria militar de los Estados Unidos. Durante esa etapa solo hubo unos pocos que tuvieron la visión de predecir el potencial mundial en aplicaciones civiles y militares. Si no fuera por las personas que sobrevivieron a esa etapa y el apoyo político y financiamiento que se les dio, este concepto no sería posible hoy en día[1].

2.2 Definiciones Claves:

Aeronave: Por lo general una aeronave es cualquier vehículo o maquina voladora en todas sus configuraciones: ala-fija, rotomotor de ala rotatoria, helicópteros, vehículos de despegue y aterrizaje vertical VTOL (vertical take-off and landing) o vehículos de despegue y aterrizaje corto STOL (small take-off and landing).

UAV: El término vehículo aéreo no tripulado (también conocido como drone) se refiere a una aeronave sin un piloto humano a bordo.

2.17 Funcionamiento y electrónica básica en aeromodelos de R/C.

El funcionamiento básico de este tipo de aeromodelos es el siguiente: mediante el uso de 4 o más servomotores se conectan varillas (Ver Fig.1) de algún material rígido (metal, aluminio o fibra de carbono) para transferir el movimiento hacia las superficies de control (alergones, elevador, timón de la dirección y acelerador del motor) con lo cual se logra el desplazamiento del mismo [25].

El control electrónico que contienen los modelos básicos consta de un transmisor y un receptor, al cual se conectan los servomotores, así como una batería. Estos equipos comerciales de R/C suelen transmitir la información de manera inalámbrica en frecuencias de 72.00 [MHz] y derivadas 72.XX [MHz] así como en 2.4 [GHz], el control de los servomotores se realiza mediante modulación por ancho de pulso, la cual explicaremos en el capítulo III.

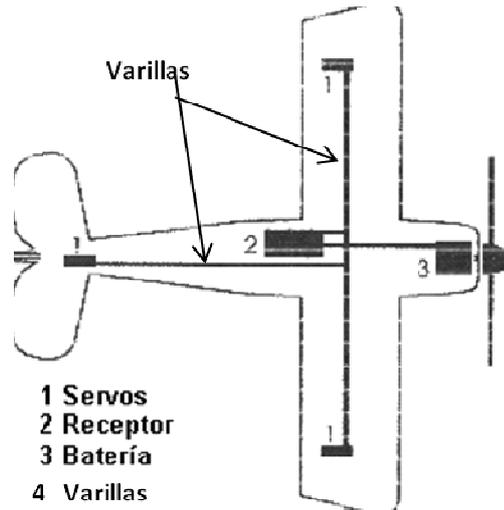


Figura 1. Funcionamiento Básico de un Avión de RC.

III Construcción e implementación del prototipo.

3.1 Introducción:

Actualmente existen en el mercado diversas tarjetas² de desarrollo para lograr la conversión de un aeromodelo de radio control a un vehículo aéreo no tripulado, la mayoría de estas plataformas son costosas o no integran los sensores requeridos para la aplicación en cuestión, sin embargo, cuentan con la ventaja de ser de código y hardware abierto para compartir información en torno al desarrollo, con diversos usuarios del mundo.

Para monitorear algunos parámetros del aeromodelo es necesario contar con algunos instrumentos de vuelo y navegación.

3.3.1 Instrumentos de Vuelo

Este tipo de instrumentos permiten conocer el estado de diversas variables durante el vuelo, los cuales pueden ser clasificados como se observa en la tabla 1:

Tabla 1 Tipos de instrumentos de vuelo.

Tipo de Instrumento	Nombre del Instrumento	Parámetros Medidos
Diferencia de presión	<ul style="list-style-type: none">- Altimetro.- Tubo de Pitot.	<ul style="list-style-type: none">- Altura Barométrica.- Velocidad Respecto del Aire.
Giroscopios	<ul style="list-style-type: none">- Horizonte artificial.	<ul style="list-style-type: none">- Ángulo de alabeo y cabeceo.
Campo magnético	<ul style="list-style-type: none">- Brújulas Magnéticas.	<ul style="list-style-type: none">- Rumbo magnético.
Auxiliares	<ul style="list-style-type: none">- Indicador de ángulo de ataque.- Cronómetros.- De nivel de líquido.- Variables climatológicas.	<ul style="list-style-type: none">- Ángulo de ataque.- Hora GMT. Tiempo Transcurrido Fecha.- Nivel de combustible.- Humedad, temperatura.

² Ej. Ardupilot Mega (Ver www.diydrones.com) o Paparazzi The Free Autopilot (ver http://paparazzi.enac.fr/wiki/Main_Page)

El GPS (Global Position System) es un sistema de navegación basado en la recepción de señales satelitales, dependiendo del tipo de GPS, este puede recibir hasta 24 señales de satélites que giran alrededor de la tierra dos veces al día, en 6 órbitas diferentes a una altura de 20 mil km. Este instrumento realiza medidas tridimensionales con una precisión aproximada, esto gracias a la tecnología empleada en los satélites y receptores. Los receptores GPS actuales nos proporcionan datos de ubicación, latitud y longitud, velocidad, hora y orientación entre otros parámetros [13].

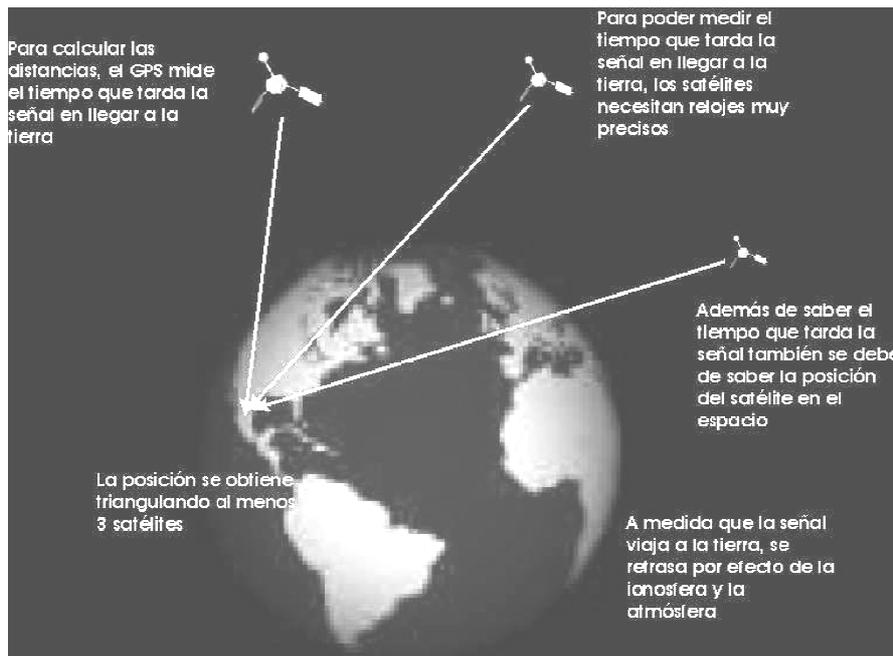


Figura 3. Funcionamiento GPS. Imagen tomada del sitio Geocienciasipc: <http://geocienciasipc.blogspot.com/2011/07/nuevas-tecnologias-sistema-de-posicion.html>

Mediante el diseño de una arquitectura escalable conseguiremos incorporar nuevos instrumentos de vuelo o navegación en el futuro, con esto se garantizan los requerimientos temporales necesarios para la adquisición de datos y el correcto control del aeromodelo, por lo que el sistema se desarrollara con dos microcontroladores (Ver Fig. 4), uno dedicado a la telemetría y otro al control de los actuadores para la estabilización del vuelo.

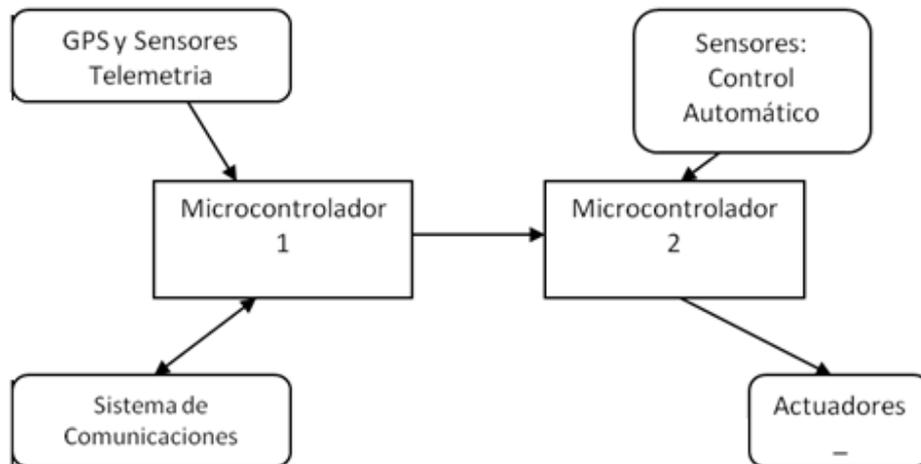


Figura 4. Diagrama básico sistema de telemetría y control automático.

3.4 Diseño e integración electrónica.

Para completar la instrumentación electrónica es necesario definir la arquitectura de la computadora de vuelo, así como el sistema de comunicaciones tierra-avión, esto nos permitirá realizar un uso eficiente de los recursos ahorrando tiempo en el desarrollo.

Nuestra computadora de vuelo debe incorporar los sensores, controladores y actuadores de manera que permitan el monitoreo y control del vehículo aéreo no tripulado. El problema principal en realizar el diseño e integración electrónica, es el peso del equipo (computadora de vuelo, baterías, transmisores, cables etc) debido a que el aeromodelo empleado cuenta con un espacio reducido en el fuselaje para albergar la electrónica.

3.5 Sensores

En esta sección describiremos brevemente los sensores:

Para la posición se hará uso de algún modulo GPS mediante el cual indicaremos las coordenadas geográficas, y la altitud, a través un altímetro barométrico.

Mediante el uso de estos dispositivos interpretaremos la posición en cualquier instante de tiempo.

Se diseñará un tubo de pitot para determinar la rapidez de nuestra aeronave, para la orientación usaremos una brújula digital indicando el rumbo que seguimos respecto del norte magnético de la tierra.

Como variables extras se podrá monitorear el estado de la temperatura exterior e interior del aeromodelo, así como la humedad relativa del ambiente.

Finalmente, para el control automático de los alerones y elevador se tiene un acelerómetro, mediante el cual se sabe la posición de la aeronave respecto a un sistema de referencia predefinido.

3.5.1 Sensor de Velocidad.

Este sensor se basa en el empleo de un tubo de pitot, el cual nos determina la rapidez del vehículo respecto del aire que lo rodea. El tubo de pitot es un instrumento con una abertura que encara al viento (dirección del avión) y otra tangencial, mediante la diferencia de presiones entre estos dos es posible calcular la rapidez del avión. A continuación se explica la teoría de funcionamiento, así como su fabricación y acondicionamiento del sensor:

Este sensor está compuesto por dos tubos concéntricos separados herméticamente (ver Fig. 5), donde el tubo interno encara al viento y es el que nos proporciona la presión dinámica: esta presión está asociada con la velocidad del fluido, mientras que el tubo externo con los orificios tangenciales es el que nos proporciona la presión estática: aquella que es independiente del fluido en movimiento [34].

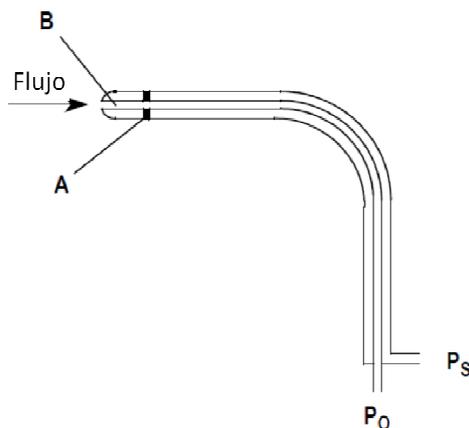


Figura. 5 Tubo de Pitot: presión dinámica en B, y presión estática en A.

Funcionamiento

Para determinar la velocidad del viento se utiliza la ecuación de Bernoulli:

$$U = \frac{P}{\rho} + \frac{V^2}{2g} + h \quad (1)$$

Donde:

U= Carga hidráulica del fluido.

P= Presión.

ρ = Densidad del aire.

h= Altura.

g= Aceleración de la gravedad.

Si aplicamos la ecuación de Bernoulli para los puntos de la figura 5 tenemos que:

$$U1 = \frac{P_o}{\rho} + \frac{V1^2}{2g} + h1 \quad (2)$$

$$U2 = \frac{P_s}{\rho} + \frac{V2^2}{2g} + h2 \quad (3)$$

Como U1 y U2 se mantienen constantes, podemos igualar 2 y 3.

$$\frac{P_s}{\rho} + \frac{V2^2}{2g} + h2 = \frac{P_o}{\rho} + \frac{V1^2}{2g} + h1 \quad (4)$$

Obteniendo así la relación de velocidad y presión en los puntos A y B, y como en los puntos anteriores podemos considerar que la aceleración, densidad, gravedad y altura son iguales se simplifica la ecuación anterior:

$$\frac{P_s}{\rho} + \frac{V_2^2}{2g} = \frac{P_o}{\rho} + \frac{V_1^2}{2g} \quad (5)$$

Como en el punto A la velocidad es cero $V=0$, debido a que el flujo de aire permanece estático, entonces:

$$\frac{V_2^2}{2g} = \frac{P_o}{\rho} - \frac{P_s}{\rho} \quad (6)$$

Por lo que mediante V_2 se tiene la rapidez del avión:

$$V_2 = \sqrt{2g \left(\frac{P_o}{\rho} - \frac{P_s}{\rho} \right)} \quad (7)$$

Si graficamos la ecuación anterior obtenemos lo siguiente [17].

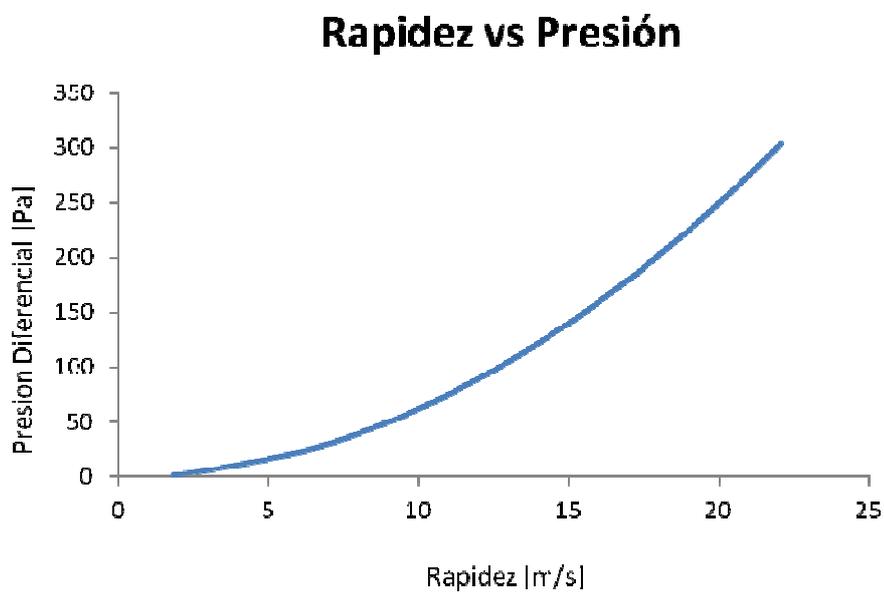


Figura 6. Relación teórica: Rapidez vs Presión diferencial en el tubo de pitot.

Fabricación:

Para la fabricación del tubo de pitot se usaron dos tubos de aluminio de diferente diámetro, tal que el mayor pudiera albergar al otro en su interior tratando de dejar un espacio entre ellos. Se utilizaron tubos de 5 [mm] de diámetro y de 2.5 [mm], ambos de 10 [cm] de longitud. Al de diámetro mayor se le realizaron orificios de 1 [mm] en los costados, mientras que el tubo interior se dejó intacto (Ver Fig. 7).



Figura 7. Tubo de Pitot-1: presión estática, entrada (A) y salida (Po). Presión dinámica, entrada (B) y salida (Ps).

Con el objetivo de obtener una óptima medición de la rapidez, se confeccionaron 3 tubos de Pitot (Ver Fig. 8), dos de ellos similares al descrito anteriormente, y un último con dos tubos separados.

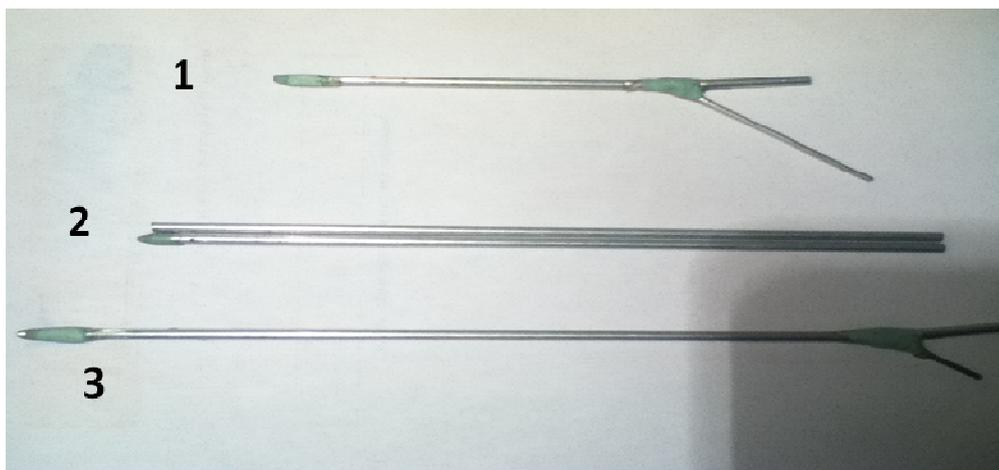


Figura 8. Tubos de Pitot fabricados.

Medición y Acondicionamiento:

Para la medición de la presión estática y dinámica se utilizó el sensor de presión diferencial MPX2010³, este tiene la ventaja de encontrarse compensado en temperatura (ver Fig.9) y puede ser conectado a un convertidor analógico digital.

Tiene un rango de 0 a 10 [Kpa] y una sensibilidad de 2.5 [mV/Kpa].

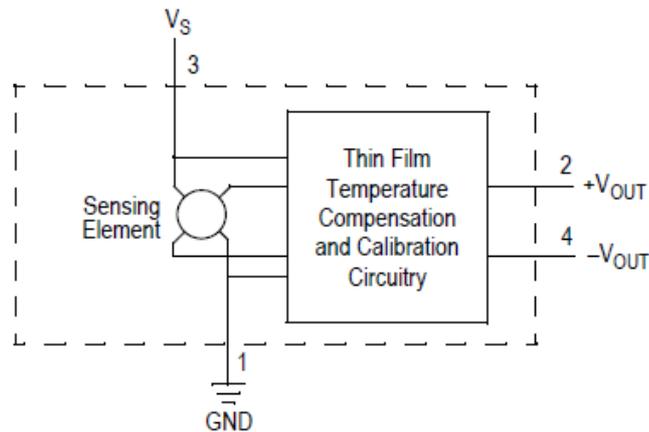


Figura 9. Diagrama de bloques del sensor de presión MPX2010.

Este tipo de sensor nos permite medir presiones diferenciales. Su tensión de alimentación típica es de 10 [V] y 16 [V] máxima y una salida de 0 a 25 [mV], tiene un comportamiento lineal regido por la siguiente ecuación:

$$V_{out} = V_{off} + sensibilidad \times P \quad (8)$$

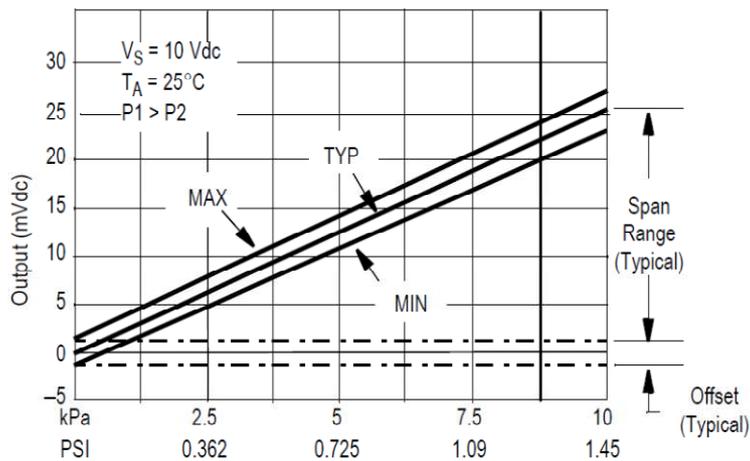


Figura 10. Salida de tensión vs Presión diferencial - Hoja de datos MPX2010.

³ De Freescale Semiconductor, la hoja de datos se puede consultar en www.freescale.com

Debido a que el microcontrolador empleado trabaja con tecnología TTL y su convertidor analógico digital es de 10 bits, se pierde resolución ya que su lectura mínima es de :

$$Resolución = \frac{5[V]}{1024[resolución\ del\ convertidor\ AD]} = 4.88 \left[\frac{mV}{bits} \right] \quad (9)$$

Por consiguiente surge la necesidad de realizar el acondicionamiento necesario a la salida de dicho sensor.

Para el acondicionamiento del sensor de presión se implementó la siguiente etapa mediante el uso de amplificadores operacionales, con una ganancia: A=500.

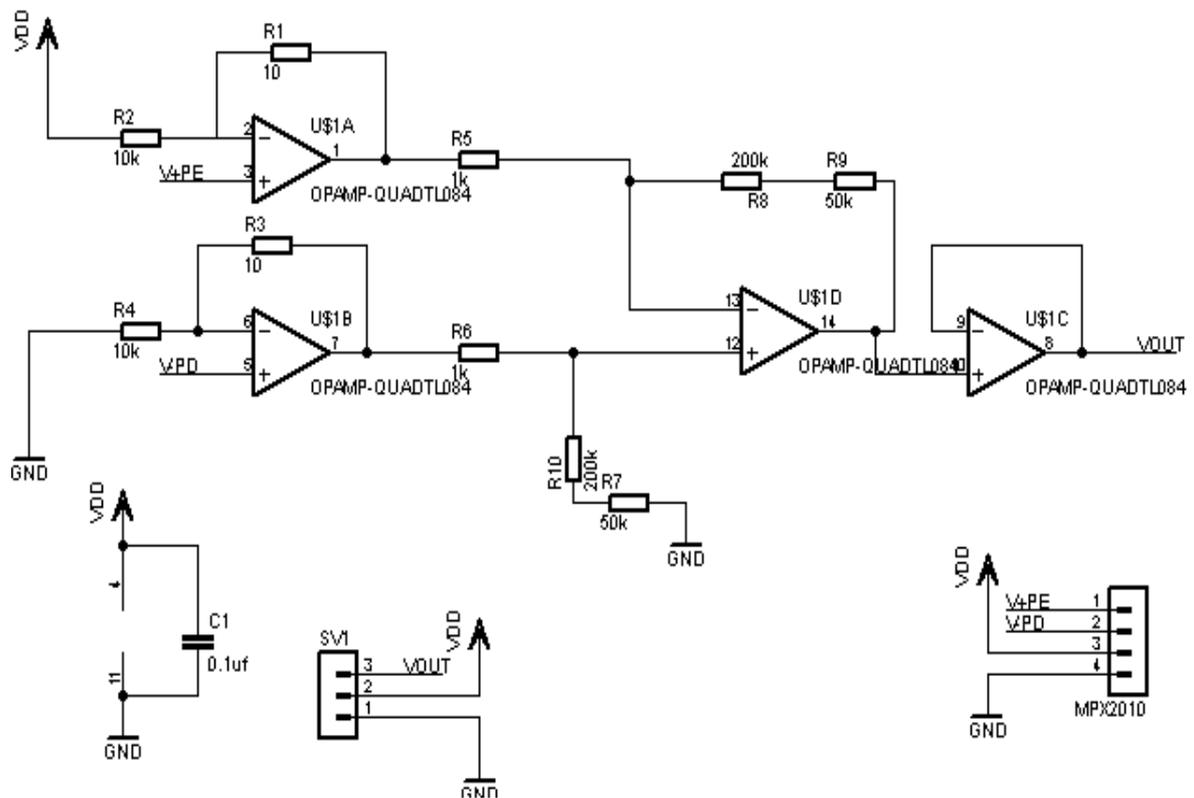


Figura 11. Circuito acondicionador para el Tubo de Pitot: Entrada de tensión de la presión estática ($V+PE$) y dinámica ($V-PD$), V_{out} = salida de presión diferencial hacia el microcontrolador.

Para evaluar el rendimiento y caracterizar cada uno de los tubos de pitot, se realizaron pruebas en el túnel de viento del laboratorio de hidráulica. Estas consistieron en asegurar en el interior del túnel cada tubo, aumentando gradualmente la velocidad del viento. Es

importante mencionar que el valor mínimo medible en el túnel es de 8 [m/s] y el máximo es de 28 [m/s] con una resolución de 0.5 [m/s].

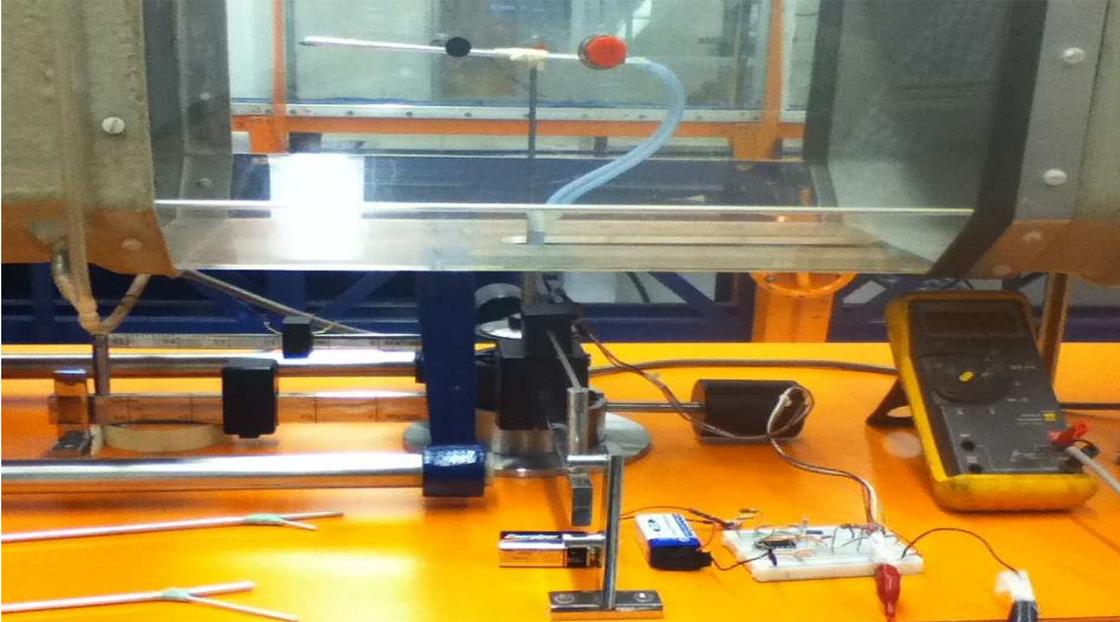


Figura 12. Banco de pruebas - Túnel de viento.

Se realizaron 3 tablas de 29 mediciones cada una, en pasos de 0.5 [m/s] para cada tubo, obteniéndose la mejor repetibilidad en el número tres, como se muestra en las siguientes gráficas.

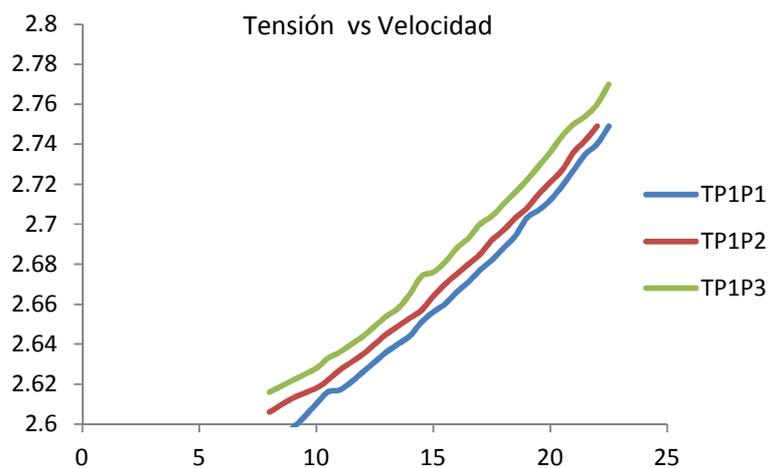


Figura 13. Resultados de caracterización -Tubo de Pitot 1.

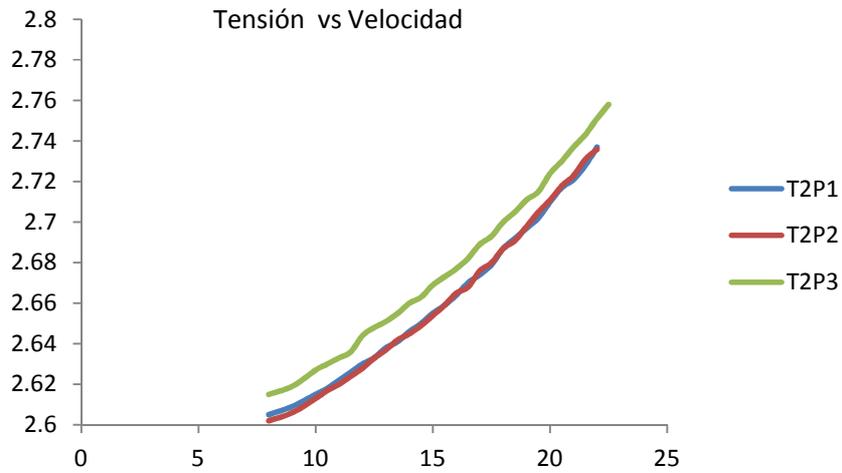


Figura 14. Resultados de caracterización -Tubo de Pitot 2.

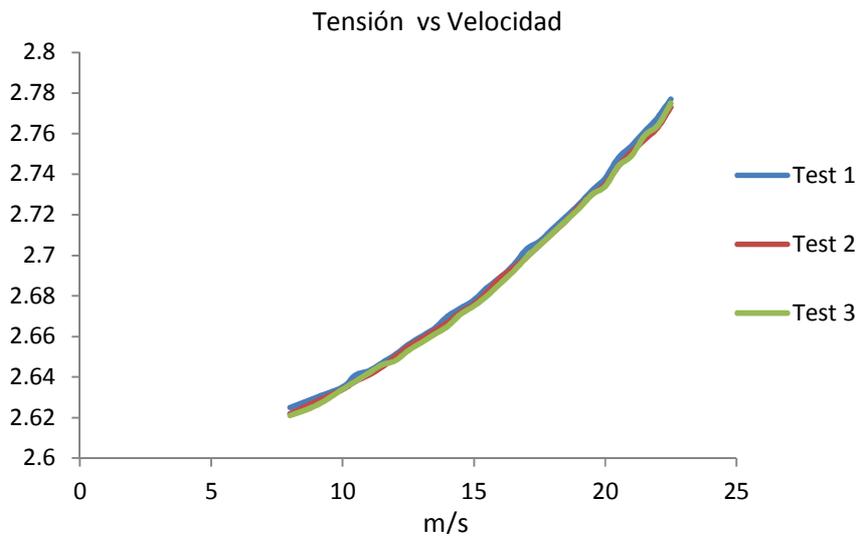


Figura 15. Resultados de caracterización -Tubo de Pitot 3.

De acuerdo a la gráfica de la figura 15 y el análisis de los datos obtenidos como se verifica en la tabla 2 concluimos que el tubo de pitot numero 3 es el de mejor desempeño.

Tabla 2. Comparación de las pruebas de caracterización para el Tubo de Pitot # 3: mediciones de rapidez y tensión.

Rapidez [m/s]	Prueba1 [V]	Prueba2[V]	Prueba3[V]
0	2.61	2.609	2.61
8	2.625	2.622	2.621
9	2.63	2.628	2.626
10	2.635	2.634	2.634
10.5	2.641	2.638	2.638
11	2.643	2.641	2.642
11.5	2.647	2.645	2.646
12	2.651	2.65	2.648
12.5	2.656	2.655	2.653
13	2.66	2.659	2.657
13.5	2.664	2.663	2.661
14	2.67	2.667	2.665
14.5	2.674	2.672	2.671
15	2.678	2.676	2.675
15.5	2.684	2.682	2.68
16	2.689	2.689	2.686
16.5	2.695	2.694	2.692
17	2.703	2.699	2.699
17.5	2.707	2.705	2.705
18	2.713	2.711	2.711
18.5	2.719	2.717	2.717
19	2.725	2.724	2.723
19.5	2.732	2.73	2.73
20	2.738	2.735	2.734
20.5	2.748	2.744	2.744
21	2.754	2.751	2.749
21.5	2.761	2.757	2.759
22	2.768	2.763	2.764
22.5	2.777	2.773	2.775

3.5.2 Sensor de Altitud

Este tipo de sensor obtiene la altitud basándose en el cambio de presión atmosférica, que se presenta en determinado punto. La dependencia de la altitud y la presión está dada por la siguiente ecuación.

$$Z = -\ln\left(\frac{p}{p_1}\right) \times R \times \frac{T}{g} \quad (10)$$

Z: Altura.

T= Temperatura promedio en grados kelvin.

P1= Presión atmosférica a nivel cero o del lugar en que actualmente se toma al medición.

P= Presión atmosférica a nivel del mar.

R= 268 constante universal de los gases.

g=9.78 aceleración de la gravedad.

Puesto que se necesita saber la presión barométrica del lugar, se ha escogido el sensor de presión MPXAZ6115A⁴ de Freescale, la sensibilidad de este dispositivo es de 45.9 [mV/kPa], funciona con una tensión de alimentación de 5 [V] y tiene una salida de 0.2 [V] a 4.8 [V], este posee la ventaja de estar compensado en temperatura y tener un comportamiento lineal regido por la siguiente ecuación:

$$V_{out} = V_s \times (.009 * P_{kpa} - 0.095) \pm Error \quad (11)$$

$$V_s = 5 \text{ [Vdc]}$$

⁴ Información para la implementación de altímetros en www.freescale.com en el documento : The Altimeter Barometer Reference Design.

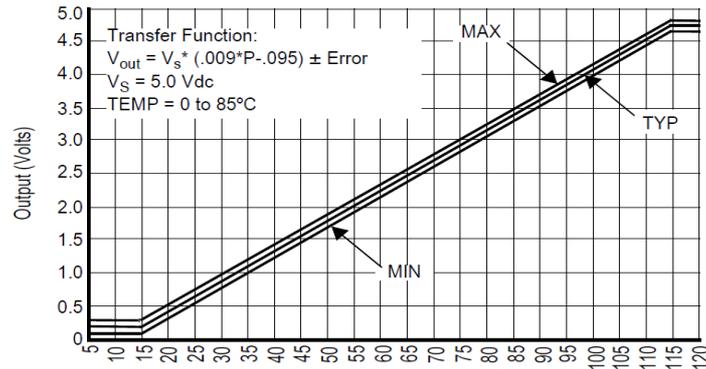


Figura 16. Tension de salida vs Presion -Hoja de Datos MPXAZ6115A.

De (11) se obtiene la expresi3n para obtener la presi3n en funci3n de la tensi3n:

$$P[kpa] = \frac{V_{out} + V_s \times 0.095}{V_s \times 0.009} \quad (12)$$

La resoluci3n de nuestro alt3metro viene determinada por la resoluci3n del convertidor anal3gico digital del μC , debido a que se esperan cambios de presi3n muy peque1os se realiza una etapa acondicionadora usando un convertidor A/D externo. Con el uso de un convertidor A/D de 16 bits sigma delta, el alt3metro tendr1 la capacidad de medir altitudes con una mayor resoluci3n.

Este tipo de convertidores son empleados en aplicaciones de audio, sensores de presi3n o aplicaciones biom3dicas, tienen una mayor precisi3n en la conversi3n a la vez que tienen un tiempo de conversi3n m1s lento, se ha escogido el uso del convertidor MCP3425⁵ de la marca Microchip en encapsulado SOT-23, este convertidor digital se comunica mediante el bus I²C ahorr1ndonos pines de entrada/salida en el μC .

⁵ Convertidor Anal3gico Digital de 16 bits con interface I2C. Ver <http://www.microchip.com>.

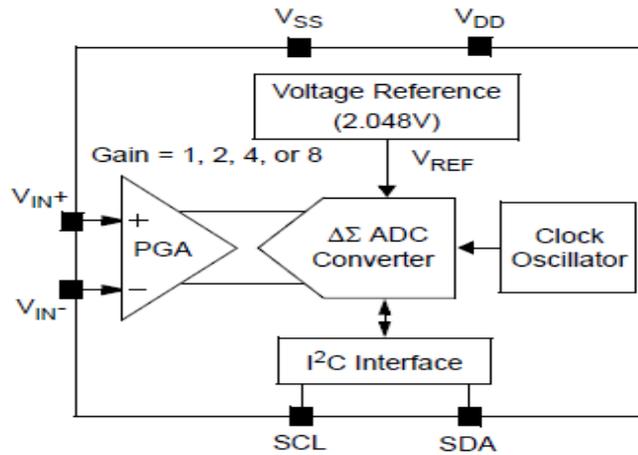


Figura 17. Diagrama de Bloques Convertidor A/D MCP3425.

Debido a que usamos un convertidor A/D externo de 16 bits se implementa la siguiente fórmula para obtener el valor de tensión en función del valor digital.

$$V_{out} = \frac{N}{65536} * V_s \quad (13)$$

N= Numero de bits 0 - 65535

Combinando (12) y (13)

$$P[kpa] = \frac{N \times V_s + V_s \times 0.095 \times 65536}{V_s \times 0.009 \times 65536} \quad (14)$$

Teniendo una medición acertada en la fuente de alimentación podemos expresar (14) en decipascales:

$$P[dpa] = \frac{N \times V_s + V_s \times 6226}{V_s \times 590} \times 100 \quad (15)$$

Usando (15) se calcula la altitud en el firmware programado en el μ C. En la figura 18 se despliega el esquema de conexiones implementado.

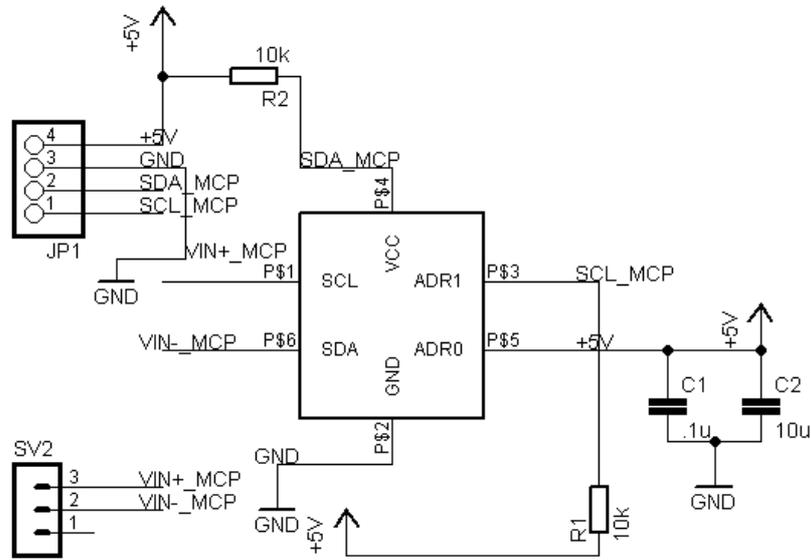


Figura 18. Esquema de conexiones empleado para el MCP34125.

3.5.3 Sensor de orientación.

Como sensor de orientación se ha usado un circuito integrado que realiza la función de brújula magnética, el modelo HMC6352⁶ de la marca Honeywell, es un integrado de solo 6x6 [mm], el cual contiene todo lo necesario para conocer la orientación respecto del polo norte magnético de la tierra. El sensor se alimenta con una fuente de 3.3 a 5 [Volts] y trabaja bajo el bus I²C, con una resolución de 0.5 grados, una vez conectado y haciendo la programación correcta mediante el μ C se obtiene el valor de orientación en grados.



Figura 19. Brújula digital modelo HMC6352 y dimensiones.

⁶ Para aplicaciones más robustas existen otras opciones, se recomienda consultar los diversos modelos en <http://www51.honeywell.com/aero/>.

El valor de orientación que nos entrega la brújula digital se procesa en el μC mediante el siguiente algoritmo.

1. Inicio de comunicación I²C.
2. Invocamos la función de lectura para el integrado HMC6352.
3. Se envía un comando para indicar la lectura del registro.
4. Se guarda el valor.
5. Finalizamos la comunicación I²C.

Posteriormente se programo al algoritmo anterior cuyo programa se muestra en el Anexo II.

3.5.4 Sensor de humedad.

Para esta variable se ha usado el sensor HMZ-433A1 (Ver fig. 20 y 21) que consiste en un circuito integrado que nos provee un voltaje proporcional a la humedad relativa del ambiente de 0 – 100%.

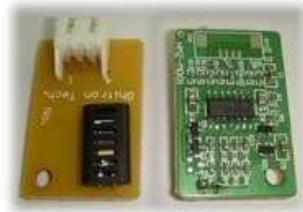


Figura 20. Sensor de Humedad HMZ-433

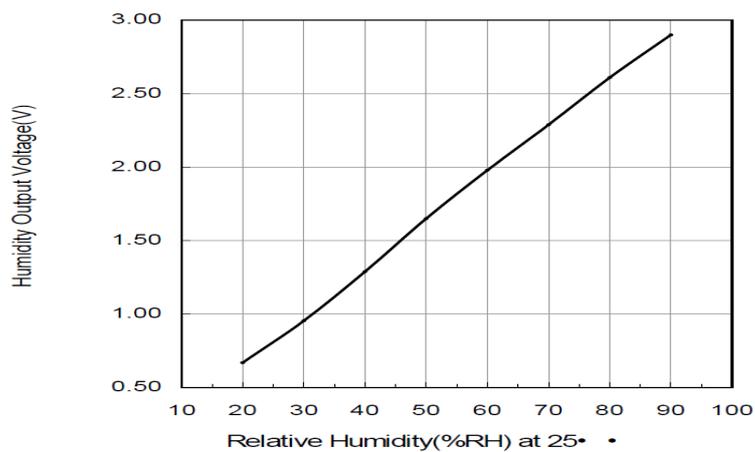


Figura 21. Salida de Tensión vs Humedad Relativa a 25°C.

La salida que nos proporciona este sensor va de 0.66 [V] a 2.94 [V] para una humedad relativa de 20% a 90%.

3.5.5 Sensor de temperatura.

Se han incorporado dos circuitos integrados uno para medir la temperatura en el exterior del avión y otro para la temperatura interior.

Medición de temperatura exterior.

Se utilizó el sensor de temperatura LM35 con encapsulado TO-92, el cual trabaja con una tensión de 4 -20 [V] , con una resolución de 10 [mV/°C] y una escala de -55 a 152 [°C], si se realiza el acondicionamiento necesario lee temperaturas inferiores a 0 [°C] lo cual no es necesario en este proyecto, este resulta útil ya que si se deseara evaluar la temperatura en diversas zonas exteriores del avión solo bastaría con adicionar mas sensores. Otra razón para incorporar este sensor es su resistencia a impactos por el tipo de encapsulado (Ver Fig.22), debido a que en la medición de esta variable no se requiere de precisión, el sensor se ha dejado conectado directamente al μC .

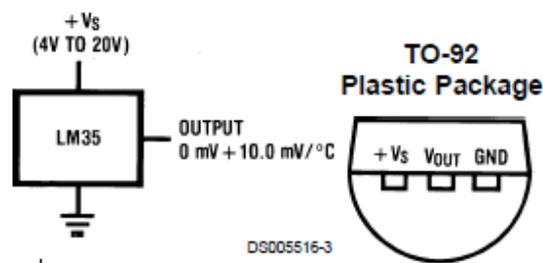


Figura 22. Diagrama de bloques y vista superior del sensor LM35.

Medición de temperatura Interior.

Con el fin de monitorear la integridad de los componentes se optó por supervisar la temperatura en el interior de la cabina, esto es muy conveniente debido a la convección que provoca la operación del motor de nuestro aeromodelo. Para realizar esta lectura se utilizó el sensor MCP9800 (ver Fig. 23) de Microchip, encapsulado SOT23, este tipo de sensor posee una precisión de ± 0.5 [°C] con 12 bits de resolución y un voltaje de operación de 2.7 a 5.5 [V], también nos da la opción de programar una alarma a determinada temperatura lo cual lo hace muy útil para nuestra aplicación.

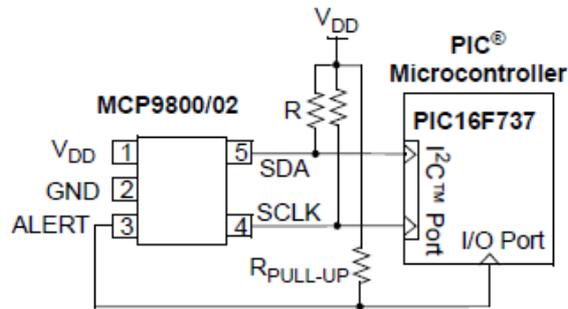


Figura 23. Interfaz de conexión recomendada MCP9800.

Para realizar la comunicación con el sensor se ha seguido el siguiente algoritmo basándose en la hoja de especificaciones.

1. Iniciamos comunicación I²C.
2. Avisamos al sensor que se requiere escribir en el registro 0x090.
3. Mandamos a escribir a los registros de interés: TA,CONFIG,THYST,TSET.
4. Mandamos a escribir la palabra de configuración.
5. Terminamos comunicación I²C.

Terminada la configuración procedemos a la lectura.

1. Iniciamos comunicación I²C.
2. Escribimos en el registro de escritura 0x090.
3. Escribimos en el registro TA 0x00.
4. Indicamos que deseamos leer el registro interno 0x091.
5. Guardamos el valor mas significativo MSB y menos significativo LSB.
6. Terminamos la comunicación I²C.

3.5.6 GPS

Para realizar la localización del aeromodelo se utilizo el modulo GPS modelo: RXM-GPS-SR⁷, este módulo contiene un receptor con antena ya integrada (Ver Fig. 24 y 25) otorgándonos la ventaja de ahorrar peso en la integración final del hardware.

⁷ Para aplicaciones más robustas se recomienda ver el catalogo de módulos GPS en: <http://www.sparkfun.com/categories/4>.

Sus características principales son las siguientes:

- Bajo consumo de potencia (46 mW).
- Alta sensibilidad (-159dBm).
- 20 Canales.
- Bajo TTFF(time to first fix) a bajas señales.
- Programación no necesaria.
- Tensión de operación 3 a 4.5 [Vdc].
- Opción para batería de respaldo.
- Interfaz UART.

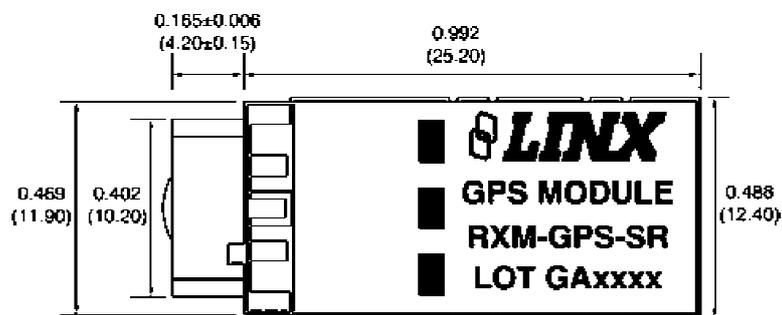


Figura 24. Dimensiones del modulo GPS [mm].

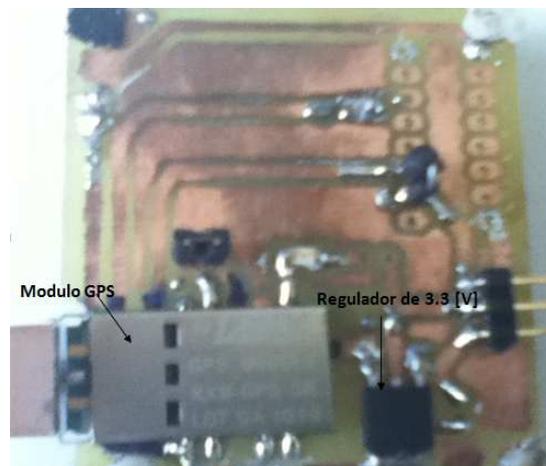


Figura 25. Tarjeta de interfaz fabricada para el modulo GPS.

Protocolo NMEA:

La salida de datos del modulo GPS está basada en el estándar NMEA-0183 (National Marine Electronics Association) el cual es muy utilizado por todos los fabricantes de GPS.

El formato de este protocolo está compuesto por los siguientes datos:

<ID-Mensaje + Cadena de datos + Checksum + Final de la Secuencia>

Para el caso particular de este módulo, la estructura de datos es enviada de forma serial a una velocidad pre-programada de 9600 bps, 8 bits de datos, 1 bit de parada, sin paridad.

Cada mensaje inicia con un carácter **\$** y termina con un retorno de carro y salto de línea **<CR><LF>**. En este formato cada campo se encuentra separado por una coma.

Un aspecto importante del mensaje es el checksum ya que este nos permite verificar la recepción de los datos, este campo se encuentra después del símbolo ***** y está compuesto de dos caracteres sin tomar en cuenta el retorno de carro y salto de línea.

El checksum consiste en la representación de dos dígitos en formato hexadecimal, este es el resultado de aplicar el operador XOR a todos los caracteres que se encuentran entre los símbolos **\$** y *****, si se llegara a encontrar algún dato vacío en la cadena, deberá descartarse debido a que todos los campos se requieren para considerar el mensaje como válido.

El módulo GPS empleado, solo nos proporciona 6 tipos de enunciados de la forma **\$GPXXX**, donde:

\$ Es el identificador de inicio del enunciado.

GP Nos identifica la procedencia del enunciado.

XXX Es el enunciado en si. Ej. RMC,etc.

El protocolo NMEA tiene 39 enunciados de los cuales solo 6 son entregados por nuestro módulo GPS. A continuación se ejemplifica y detallan los valores de una cadena:

\$GPRMC,053740.000,A,2503.6319,N,12136.0099,E,2.69,79.65,100106,,,A*53

Tabla 3. Valores y descripción del enunciado GPRMC.

Nombre	Ejemplo	Unidades	Descripción
ID-Mensaje	\$GPRMC		Identificador del protocolo
Hora UTC	53740		hhmmss.sss
Estado	A		A=dato valido ó V=dato no valido
Latitud	2503.632		ddmm.mmmm
Indicador N/S	N		N=norte ó S=sur
Longitud	12136.01		ddmm.mmmm
Indicador E/W	E		E=este ó W=Oeste
Velocidad sobre Tierra	2.69	Nudos	Verdadero
Curso Sobre Tierra	79.65	Grados	
Fecha	100106		ddmmyy
Variación Magnética			
Variación de Censado			E=este ó W=oeste(no se muestra)
Modo	A		A= Autónomo D=DGPS, E=DR
Checksum	*53		
<CR><LF>			Terminación final del mensaje

Rutina para la lectura del módulo GPS.

Para la lectura de datos se ha empleado el siguiente algoritmo:

1. Inicializar la interfaz serial de datos del módulo GPS.
2. Guardar los datos del formato NMEA en un buffer de recepción.
3. Analizar los datos del mensaje usando las comas como separador.
4. Actualizar los datos de interés en nuestra aplicación.
5. Limpiar el buffer de recepción y dejarlo listo para la siguiente recepción de datos.
6. Verificar el mensaje mediante el cheksum y el campo de dato válido (**A**).
7. Transmitir el mensaje y regresar al paso 1.

Cuyo diagrama de flujo detallamos en la siguiente figura:

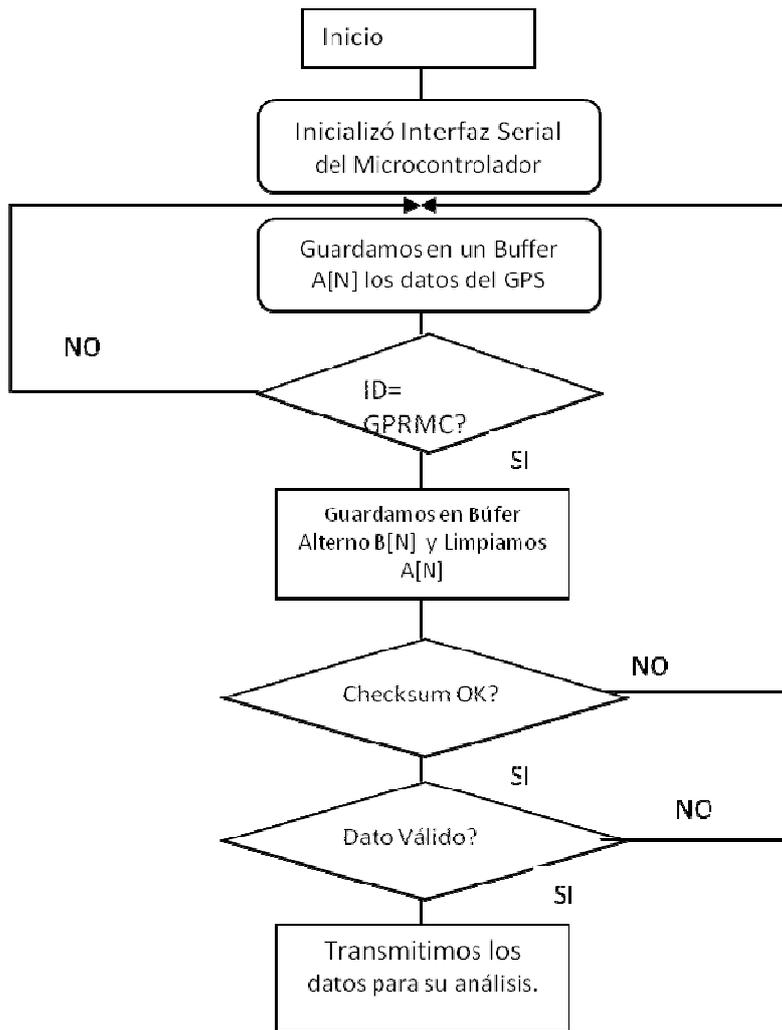


Figura 26. Diagrama de flujo para la lectura del GPS.

3.6 Implementación de la computadora de vuelo o FMS (flight management system).

El sistema de gestión de vuelo se designa con el acrónimo FMS [23], que en esencia es la unión de un sistema de telemetría y un piloto automático.

Como se mencionó en 3.3.2 se ha escogido el uso de dos microcontroladores, separando la telemetría y el control automático en dos tarjetas, mediante esta arquitectura tenemos la ventaja de dedicar un μC exclusivamente para la lectura de sensores, monitoreo de la navegación así como transmisión y recepción de datos, dejando al otro únicamente el control de las superficies (alergones, elevador y timón de la dirección), en cuanto al control

del empuje, esté se manipula desde la emisora de R/C original debido al riesgo que implica un paro de motor, pues solo puede encenderse en tierra.

El Diagrama de bloques propuesto para la arquitectura descrita es el siguiente:

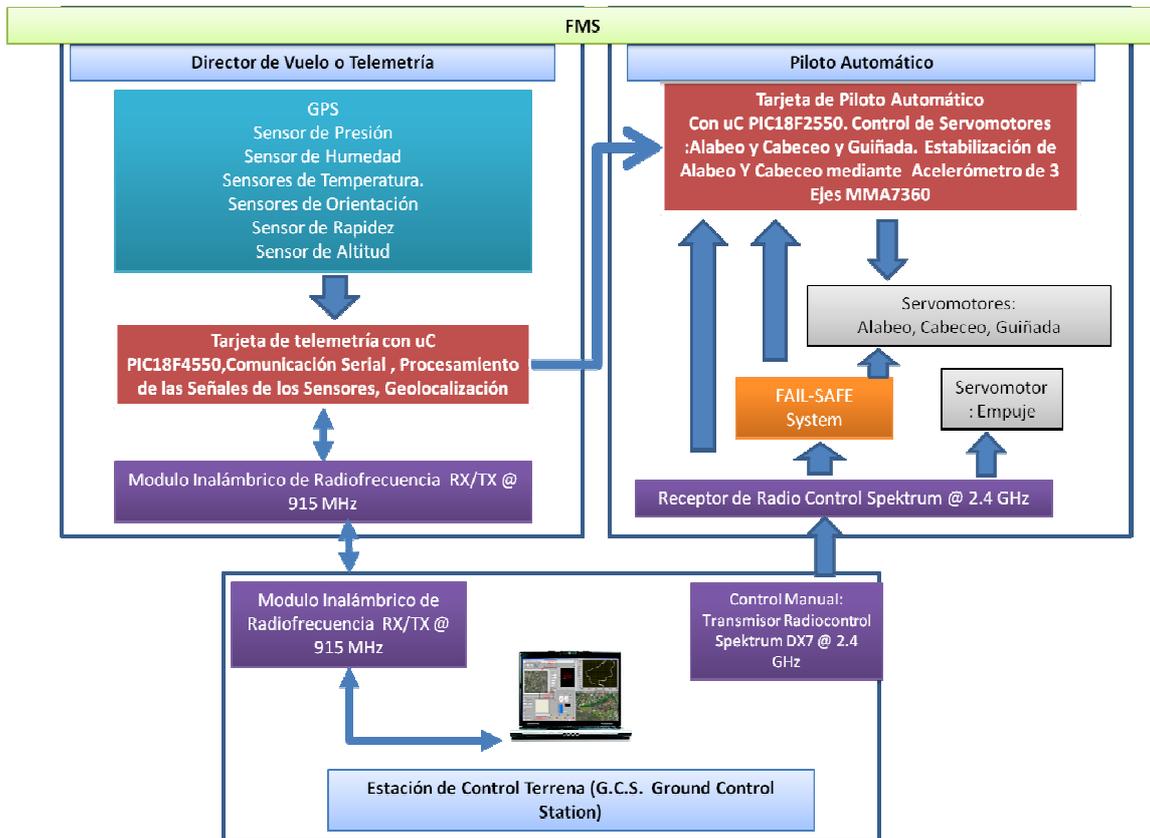


Figura 27. Diagrama de bloques del sistema UAV propuesto.

3.6.1 Descripción de la computadora de vuelo.

En la figura 27 observamos en color rojo el núcleo de cada sección, para la telemetría y el piloto automático se utiliza en PIC18F4550 y 2550 respectivamente. En la sección de telemetría resaltado en color azul turquesa se encuentran incorporados los sensores de presión, humedad, temperatura, orientación, rapidez y altitud así como el modulo GPS, esta sección se encarga de recabar el estado de los datos de interés y enviarlos a la estación terrena mediante un modulo inalámbrico de comunicaciones (sección en color morado), a su vez la estación terrena puede enviar comandos a la computadora de vuelo, la cual tiene la capacidad de activar o desactivar el piloto automático.

La sección del piloto automático se encarga de controlar los movimientos de alabeo y cabeceo usando como retroalimentación la lectura de un acelerómetro de 3 ejes, esta sección contiene un multiplexor mediante el cual se alterna entre control automático o manual en caso de alguna contingencia. La conmutación es controlada mediante un μC de gama baja conectado al receptor del radiocontrol, este último es totalmente independiente en cuanto a su fuente de alimentación y responde únicamente al radiocontrol comercial anulando toda acción del piloto automático. Esta última sección es nombrada como Fail Safe System (sección de color naranja).

Finalmente en la sección inferior tenemos la Estación de Control Terrena (G.C.S.)⁸ compuesta por otro módulo inalámbrico de comunicaciones conectado a la computadora, en esta se lleva a cabo el procesamiento de datos mediante lo cual es posible monitorear el estado de los sensores así como saber la ubicación y orientación del avión mediante indicadores virtuales y un mapa dinámico.

3.6.2 Tarjeta de telemetría.

El desarrollo de la tarjeta de telemetría incluye un μC PIC18F4550, su programación se realizó en lenguaje C⁹, este fue elegido por los siguientes parámetros y características.

- Rendimiento de 48[MHz] (12 MIPS).
- 13 Canales A/D de 10 Bits.
- Periféricos de comunicación digital: USART, I²C.
- Memoria de programa de 32 KB.
- RAM 2048 bytes.
- Temporizador de 8 y 16 Bits.
- 35 Puertos de entradas/salida.

Basándonos en estas características, la figura 27 y los instrumentos de vuelo desarrollados se procedió a realizar una serie de pruebas y validaciones a los sensores en conjunto con el μC , logrando una correcta comunicación entre estas.

Para las pruebas iniciales se utilizó un μC en encapsulado DIP40 ya que es fácil de utilizar en una tarjeta de prototipos (ver Fig. 28).

⁸ Si no se quisiera desarrollar el software para la estación de control terrena en la red existen diversos desarrollos de código abierto para no partir desde cero, como: Openpilot G.C.S., QgroundControl, PixHawk G.C.S., YAGCS etc.

⁹ Para realizar la programación se utilizó el compilador CCS C : www.ccsinfo.com.

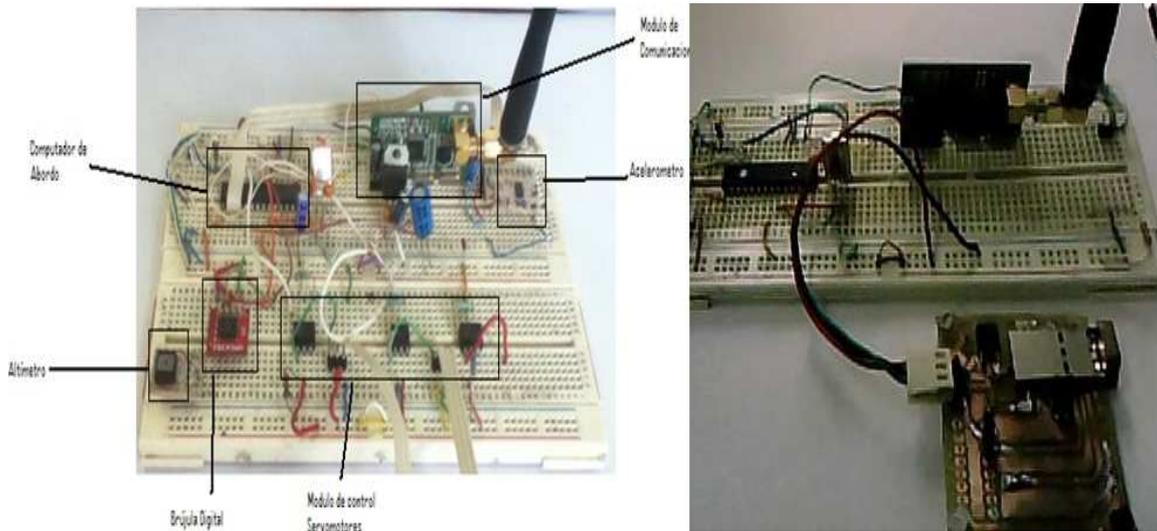


Figura 28. Realización de pruebas: Prototipo tarjeta de telemetría y GPS.

Los pasos a seguir para verificar esta etapa fueron los siguientes:

- Conexión de los sensores con el μC en una tarjeta de prototipo.
- Programación del firmware del μC .
- Recepción de datos del modulo GPS y envió a una terminal virtual RS-232.
- Envío de datos de los sensores a una terminal virtual RS-232.
- Depuración del firmware del μC .

Realizadas las acciones anteriores se procedió a enviar la información a través del modulo de comunicaciones inalámbricas para verificar que se recibían correctamente los datos (ver Figuras 29 a 32).

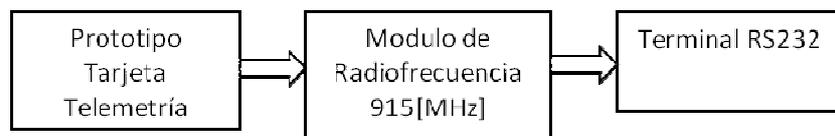


Figura 29. Diagrama de bloques para el envío de información.

Inicialmente se verificó la lectura del GPS, como se observa en la figura 30, se logró una correcta comunicación con este modulo logrando visualizar las 6 diferentes cadenas de datos: GPRMC, GPVTG, GPGGA, GPGLL, GPGSA, GPGSV. De estas sentencias solo se utilizo la RMC ya que contiene los datos de latitud y longitud.

```

Virtual Terminal - GPS
$GPRMC,224553.720,A,1919.2584,N,09911.3046,W,39.16,172.20,200910,0.0,A
$GPVTG,172.20,T,M,39.16,N,72.5,K,A*06
$GPGGA,224554.720,1919.2479,N,09911.3040,W,1.05,1.8,2299.1,M,-9.0,M,
$GPGLL,1919.2479,N,09911.3040,W,224554.720,A,A*41
$GPGSA,A,3,02,10,05,17,04,,,,,,,,,3.6,1.8,3.1*38
$GPGSV,3,1,11,02,64,327,25,10,56,118,25,04,40,026,26,05,36,180,19*76
$GPGSV,3,2,11,17,19,075,19,12,35,325,22,09,30,250,20,27,29,238,*75
$GPGSV,3,3,11,28,10,139,21,25,03,322,,30,,322,*4D
$GPRMC,224554.720,A,1919.2479,N,07811.3066,W,38.59,173.47,200910,0.0,A
$GPVTG,173.47,T,M,38.59,N,71.5,K,A*0F
$GPGSV,3,2,11,17,19,075,18,12,35,325,21,09,30,250,20,27,29,238,*77
$GPGSV,3,3,11,28,10,139,23,25,03,322,,30,,322,*4F
$GPRMC,224606.000,A,1919.1449,N,09911.2886,W,31.43,169.07,200910,0.0,A
$GPVTG,169.07,T,M,31.43,N,58.2,K,A*0E
$GPGGA,224607.000,1919.1360,N,09911.2869,W,1.05,1.8,2324.1,M,-9.0,M,
$GPGLL,1919.1360,N,09911.2869,W,224607.000,A,A*4F
$GPGSA,A,3,02,10,05,17,04,,,,,,,,,3.6,1.8,3.1*38
$GPGSV,3,1,11,02,64,327,24,10,56,118,24,04,40,026,27,05,36,180,16*78
$GPGSV,3,2,11,17,19,075,18,12,35,325,21,09,30,250,21,27,29,238,*76
$GPGSV,3,3,11,28,10,139,23,25,03,322,,30,,322,*4F
$GPRMC,224607.000,A,1919.1449,N,09911.2886,W,31.43,169.07,200910,0.0,A

```

Figura 30. Sentencias de datos entregadas por el modulo GPS-RXM-SR.

Una vez verificado que el envío de datos fue correcto se procedió a programar una rutina que únicamente nos enviara la sentencia GPRMC y a su vez comprobara el checksum como se aprecia en la figura 31.

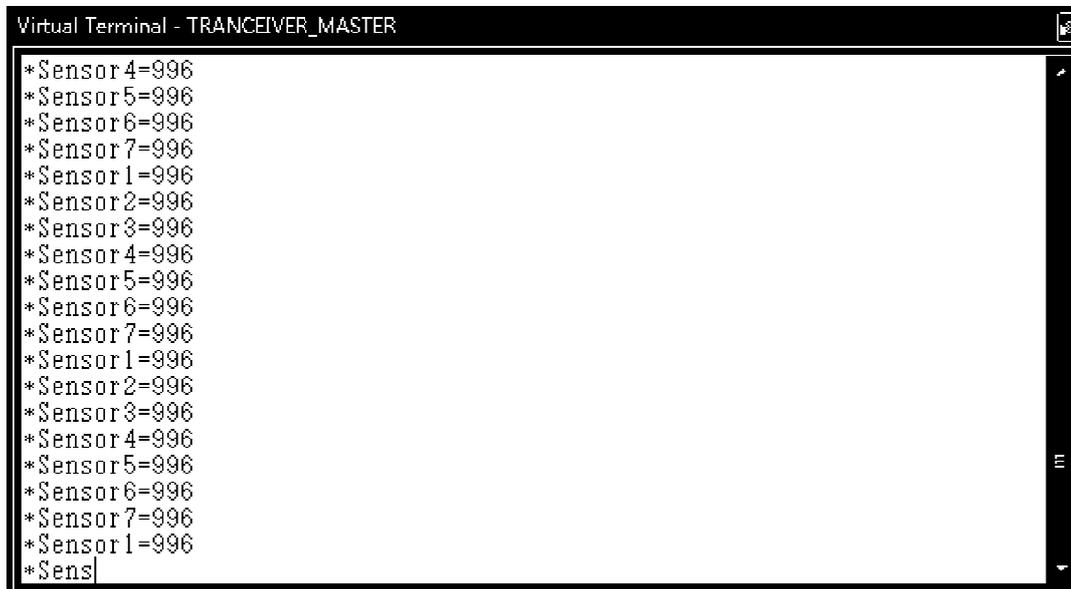
```

Virtual Terminal - TRANCEIVER_MASTER
--checksum=46--
*error=0
$GPRMC,224554.720,A,1919.2479,N,07811.3066,W,38.59,173.47,200910,0.0,A*4D
--checksum=42--
*error=1
$GPRMC,224606.000,A,1919.1449,N,09911.2886,W,31.43,169.07,200910,0.0,A*42
--checksum=42--
*error=1
$GPRMC,224607.000,A,1919.1360,N,09911.2869,W,32.24,171.22,200910,0.0,A*42
--checksum=4B--
*error=1
$GPRMC,224615.000,A,1919.0611,N,09911.2769,W,32.60,170.06,200910,0.0,A*4B
--checksum=4E--
*error=1
$GPRMC,224616.000,A,1919.0524,N,09911.2752,W,31.11,168.76,200910,0.0,A*4E
--checksum=4D--
*error=1
$GPRMC,224554.720,A,1919.2479,N,09911.3040,W,38.59,173.47,200910,0.0,A*4D
*c0n706*c1n706*c2n706*c3n706*hmc6553.5*

```

Figura 31. Se muestra el envío de la sentencia GPRMC y la verificación de checksum enmarcadas por líneas.

Finalmente se enviaron datos de los diversos sensores a la terminal virtual visualizándose lo siguiente:



```
Virtual Terminal - TRANCEIVER_MASTER
*Sensor 4=996
*Sensor 5=996
*Sensor 6=996
*Sensor 7=996
*Sensor 1=996
*Sensor 2=996
*Sensor 3=996
*Sensor 4=996
*Sensor 5=996
*Sensor 6=996
*Sensor 7=996
*Sensor 1=996
*Sensor 2=996
*Sensor 3=996
*Sensor 4=996
*Sensor 5=996
*Sensor 6=996
*Sensor 7=996
*Sensor 1=996
*Sens|
```

Figura 32. Envío de valores de los sensores.

3.6.3 Definiendo reglas de comunicación

Como se aprecia en la figura 32 el envío de datos de nuestros sensores no es el adecuado, este no contiene ningún formato que nos permita identificar fácilmente entre estos realizar un procesamiento posterior, por lo cual se decidió implementar un formato para la separación de datos.

Teniendo presente que tenemos 6 distintas variables: presión, temperatura 1, temperatura 2, humedad, rapidez, altitud y que nuestra plataforma está abierta a incorporar mas dispositivos según sean necesarios, se propone enviar en una única sentencia de datos el valor de los sensores separados por diferentes identificadores, ej.

$$* CX_0Nvalor_0 * CX_1Nvalor_1 * CX_2Nvalor_2 * CX_3Nvalor_3 \dots * CX_nNvalor_n *$$

Donde CX_n es el número de sensor y $valor_n$ es el dato, por lo que aplicando esto se crea la siguiente tabla:

Tabla 4. Descripción de los identificadores.

Identificador	Variable	Unidad
CX ₀ N	Presión	[kPA]
CX ₁ N	Temperatura 1	[°C]
CX ₂ N	Temperatura 2	[°C]
CX ₃ N	Humedad	[% RH]
CX ₄ N	Rapidez	[m/s]
CX ₅ N	Altitud	[m]

Realizado lo anterior, se procedió al diseño del diagrama esquemático (Anexo I), y su fabricación en placa de circuito impreso (ver Fig.33 y 34).

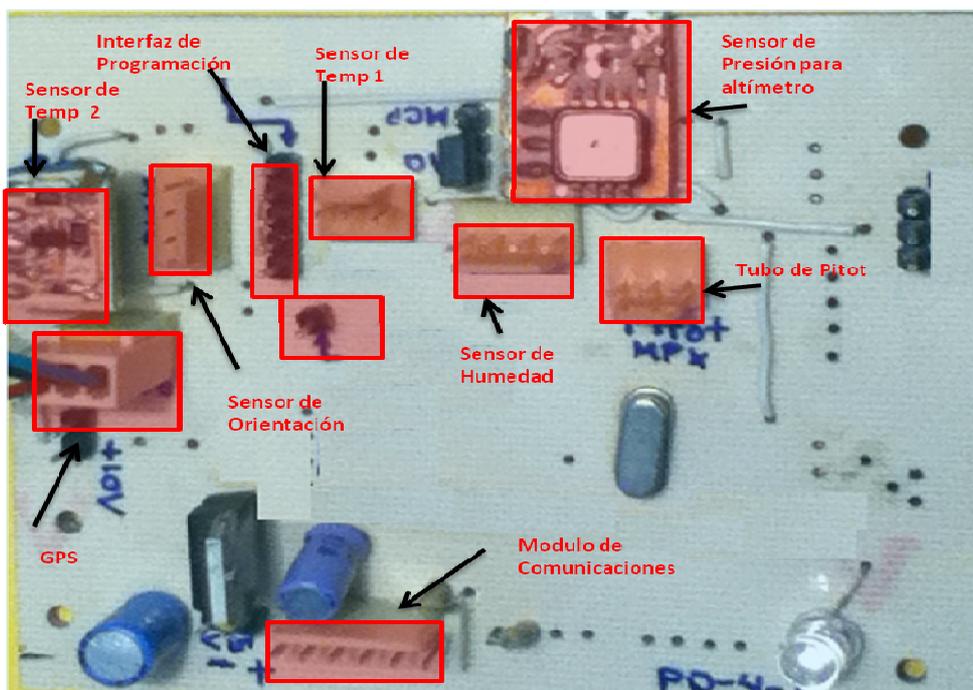


Figura 33. Cara superior tarjeta de telemetría.

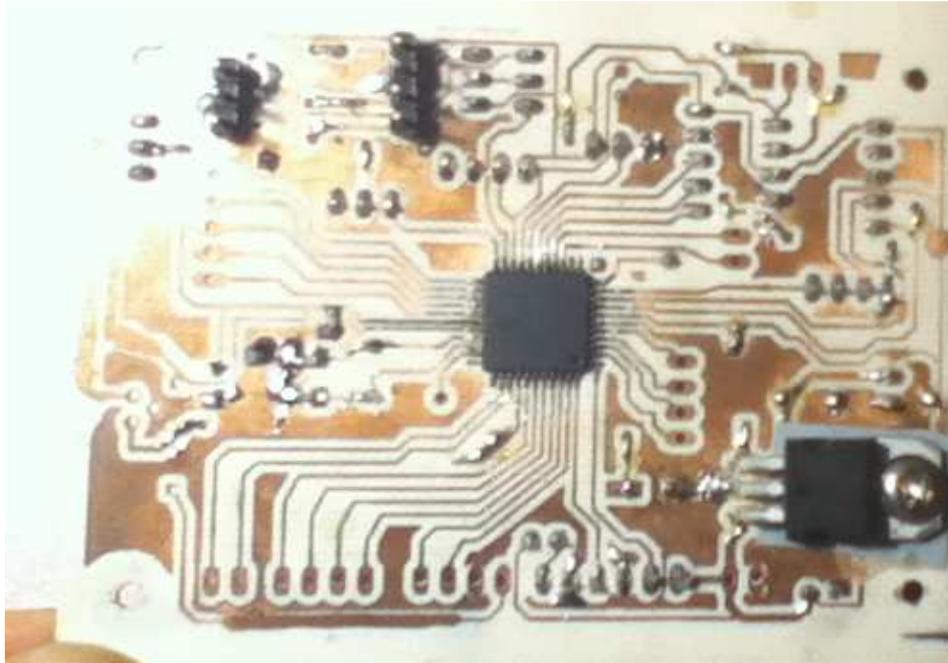


Figura 34. Cara inferior tarjeta de telemetría.

Concluida la fabricación del sistema de telemetría se planteó el siguiente diagrama de flujo (ver Fig. 35) para la operación del sistema, cuya programación se muestra en el Anexo II.

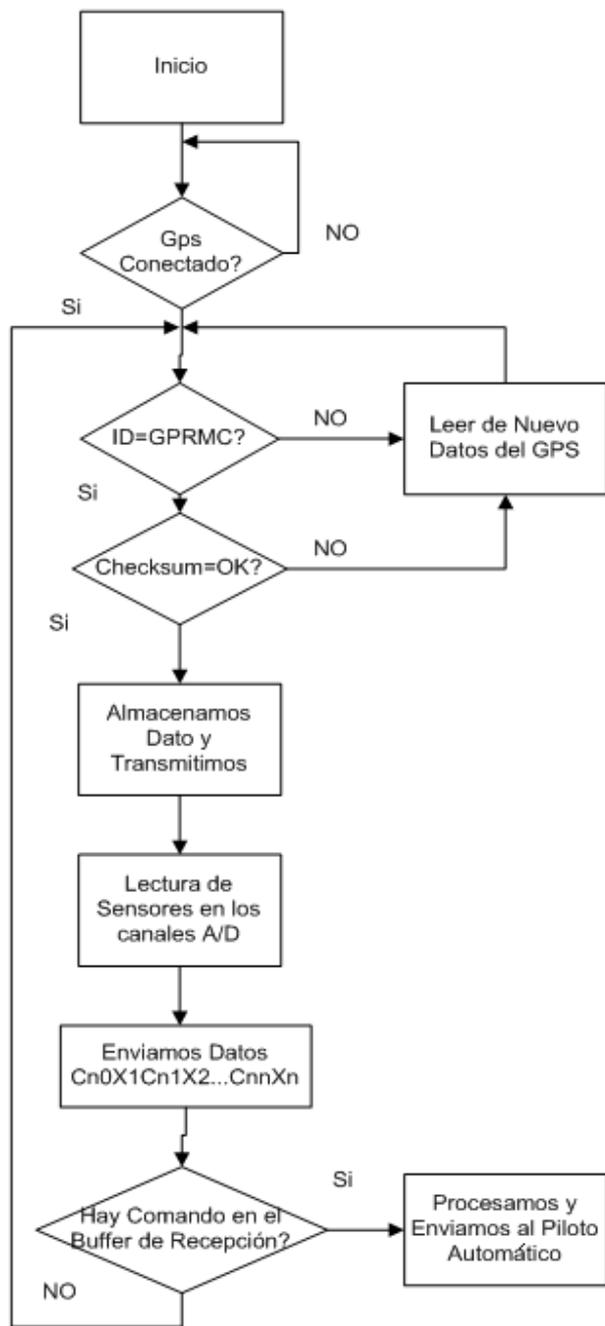


Figura 35. Diagrama de flujo sistema de telemetría.

3.6.4 Piloto automático.

Para el piloto automático¹⁰ se eligió el μ C PIC18F2550 el cual es una versión similar al PIC18F4550 en rendimiento, con la excepción de que éste contiene menos periféricos como se detalla a continuación.

- Rendimiento de 48 MHz(12 MIPS).
- 10 Canales A/D de 10 Bits.
- Periféricos de comunicación digital: USART,I2C.
- Memoria de programa de 32 KB.
- RAM 2048 bytes.
- Temporizador de 8 y 16 Bits.
- 24 Puertos de entrada/salida.

Debido a que las superficies de control en los aeromodelos son manipuladas mediante el uso de servomotores es necesario entender su funcionamiento así como realizar un algoritmo que permita manipular simultáneamente y de manera independiente cada uno, en esta etapa caemos en la dificultad de solamente contar con dos módulos PWM en el μ C por lo cual para realizar el control simultáneo se ha decidido implementar el control por software.

3.6.5 Teoría para el control de servomotores.

Un servomotor es un dispositivo que se manipula mediante una señal de control PWM (modulación por ancho de pulso), esto se logra mediante un tren de pulsos con diferentes valores para el tiempo en bajo y alto con valores de 0 [V] y 5[V] respectivamente, así como un mismo periodo.

Con esta señal se manipula el movimiento, dependiendo del valor que contenga, la electrónica interna se encarga de colocar al motor en una posición particular dependiendo del ciclo de trabajo.

Generalmente estos trabajan a una frecuencia de 50[Hz] (ver Fig. 36) o un periodo $T= 20$ [ms] y su rango de movimiento es de 180 grados, por ejemplo para un valor de 0.7 [ms] el servomotor se posicionaría en -90° , y con un valor de 2.3 [ms] en $+90^\circ$, esto significa que

¹⁰ Para robustecer un piloto automático se recomienda usar una Unidad de Navegación Inercial (Visitar <http://www.sparkfun.com/categories/160>).

haciendo variar el ciclo de trabajo en este intervalo se posiciona el servomotor en una posición de 0° a 180°.

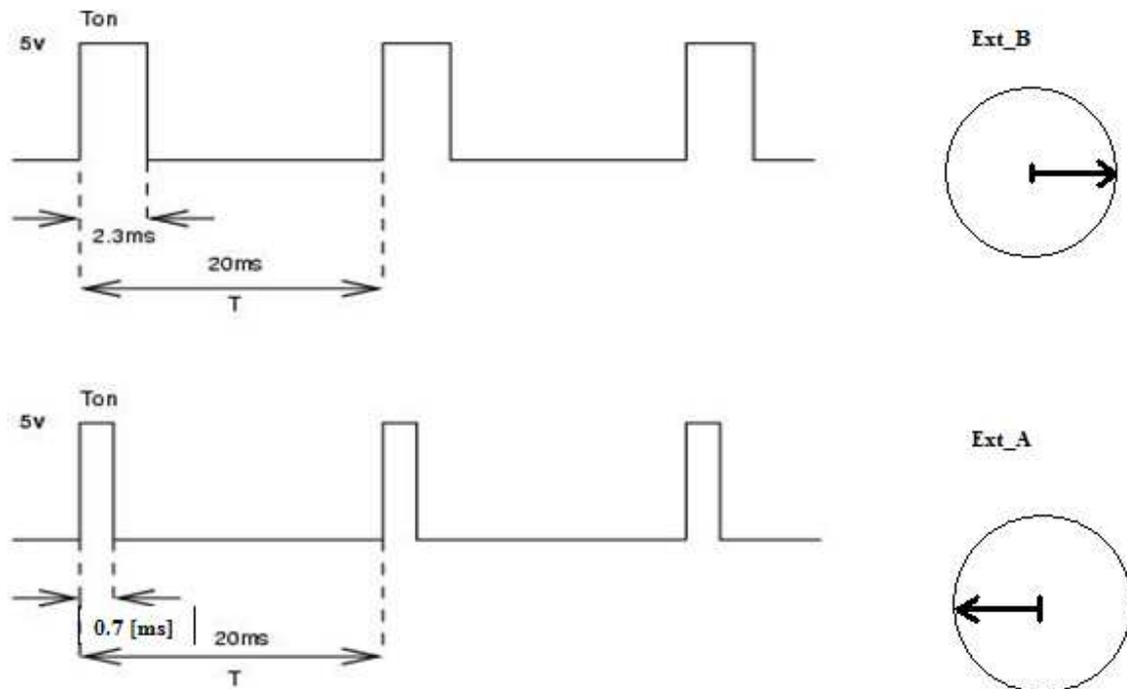


Figura 36. Señal PWM para el control de un servomotor.

3.6.6 Método para el control de 3 o más servomotores.

El problema a resolver es controlar más de un servomotor de manera simultánea, sabemos que el periodo de nuestra señal de control es $T=20$ [ms], para nuestro caso hemos decidido dividir este periodo entre 4 lo cual nos otorga una ventana de tiempo para controlar a cada uno de los servomotores:

$$\text{Ventana} = \frac{20}{4} = 5[\text{ms}] \quad (16)$$

Esto significa que para controlar cada servomotor tendremos una ventana de 5 [ms] (ver Fig. 37) de duración pero de manera consecutiva hasta completar el control de los 4. Debemos de recordar que ninguno debe superar los 2.3 [ms] puesto que fuera de esos límites el servomotor no responderá.

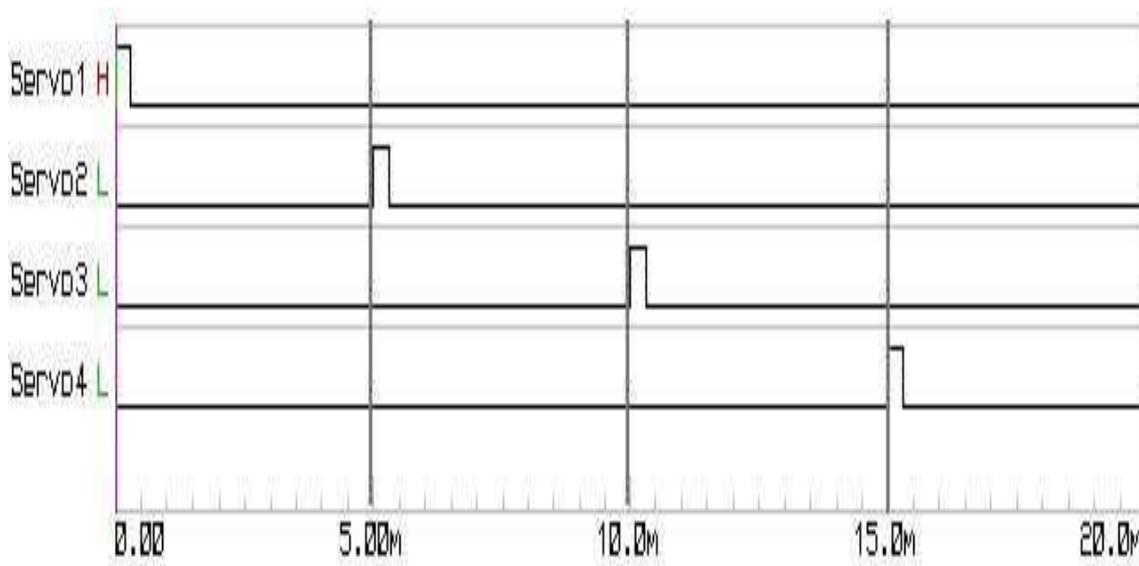


Figura 37. Ventana de tiempo para el control de 4 servomotores.

Dentro de cada ventana de 5[ms] se pone en alto y bajo la señal de control de un solo servomotor, sumando las 4 ventanas a cada uno le corresponderá estar en alto de nuevo a los 20 [ms]. Por lo cual cada uno estará trabajando a una frecuencia de 50[Hz] independientemente del valor en alto que tenga cada uno.

3.6.7 Control automático de alerones y estabilizador horizontal.

Para tener de manera automática el movimiento de las superficies de control, no solo basta mover los servomotores, en este caso es necesario saber el ángulo de ataque en el que se encuentra el aeromodelo, así como el ángulo de alabeo, y basados en datos realimentar los servomotores para estabilizarlo de manera automática. Se ha decidido usar un acelerómetro de 3 ejes XYZ, con este obtenemos el ángulo de inclinación en los tres ejes. Se utilizó el modelo MMA7360 de la marca Freescale.

El principio de funcionamiento de estos dispositivos se logra con la disposición de dos celdas capacitivas micromaquinadas orientadas de acuerdo a un sistema de ejes cartesiano XYZ, estas varían su distancia produciendo un cambio en el valor de capacitancia, de acuerdo como se ejemplifica en la figura 38.

Aceleración →

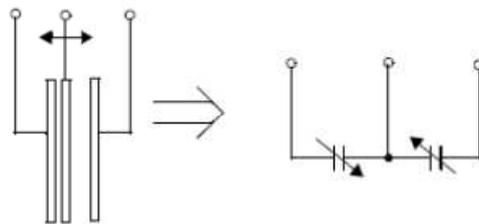


Figura 38 . Funcionamiento básico del acelerómetro.

Los parámetros principales del dispositivo son los siguientes.

Tabla 5. Parámetros del acelerómetro MMA7360.

Tensión de operación:	2.2-3.6 [V]
Sensibilidad	+/- 1.5[g] y +/- 6[g]
Resolución	800[mV/g]@1.5 [g] y 206[mV/g]@6[g]
Detección de caída libre	0[g]

El diagrama de conexiones recomendado por el fabricante es el siguiente:

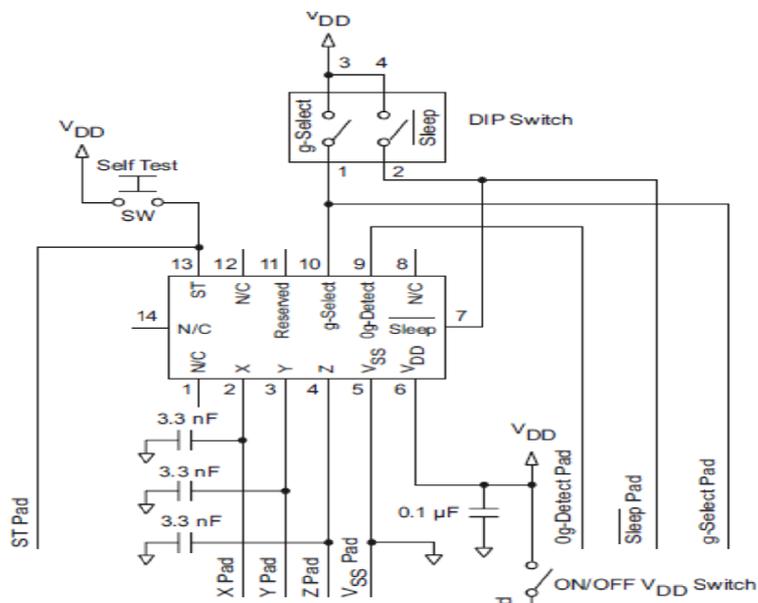


Figura 39. Diagrama de conexiones del acelerómetro MMA7360.

Asimilado el funcionamiento de este dispositivo es necesario saber de qué manera se obtiene el valor del ángulo respecto a un sistema de ejes XYZ, para esto se ha consultado la nota de aplicación de Freescale AN3461.

Medición de los ángulos de inclinación de roll, pitch y yaw.

De acuerdo a la nota de aplicación AN3461, si quisiéramos saber el ángulo de inclinación respecto a los tres ejes debemos usar las tres salidas analógicas del acelerómetro de acuerdo a lo siguiente.

La letra ρ (Ang pitch) está definido como el ángulo del eje Y relativo a la tierra.

La letra θ (Ang Roll) está definido como el ángulo del eje X relativo a la tierra.

La letra ϕ (Ang Yaw) está definido como el ángulo del eje Z relativo a la tierra.

Para obtener el valor de aceleración en los 3 ejes tenemos que:

$$Ax = \frac{Voutx - Voff}{S} \quad (17)$$

$$Ay = \frac{Vouty - Voff}{S} \quad (18)$$

$$Az = \frac{Voutz - Voff}{S} \quad (19)$$

Donde **Vout** es el valor de tensión proporcionado por su respectivo eje(x, y ó z).

Voff es el valor de tensión de offset y **S** es el valor de sensibilidad del acelerómetro.

Haciendo uso de los valores de (17),(18) y (19) se calcula el valor del ángulo de inclinación(ver Fig. 40):

$$\theta = \tan^{-1} \left(\frac{Ax}{\sqrt{Ay^2 + Az^2}} \right) \quad (20)$$

$$\rho = \tan^{-1} \left(\frac{Ay}{\sqrt{Ax^2 + Az^2}} \right) \quad (21)$$

$$\varphi = \tan^{-1} \left(\frac{\sqrt{Ax^2 + Ay^2}}{Az} \right) \quad (22)$$

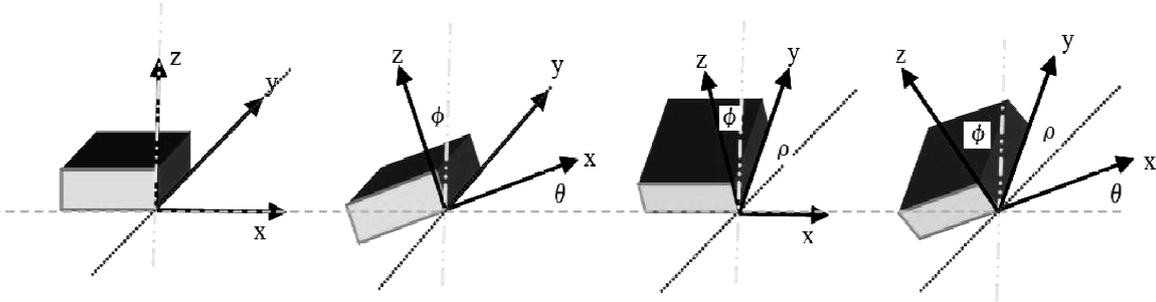


Figura 40. Ángulos de Roll, Pitch y Yaw mediante el acelerómetro.

Programando las ecuaciones (17),(18) ,(19) y (20),(21), (22) se obtiene el valor del ángulo de inclinación ya sea en el μC o en nuestra interfaz gráfica.

3.6.8 Control PID de alabeo y cabeceo.

El control PID (proporcional integral y derivativo) es un método de control con realimentación que se utiliza comúnmente en la industria, este es fácilmente adaptable a sistemas lineales o no lineales.

El control PID se basa en calcular el error entre un valor medido y el que se quiere obtener tal que podamos aplicar una acción para ajustar dicho valor.

La acción de control proporcional nos determina la corrección al error actual.

La acción integral genera una corrección al error estacionario.

La acción derivativa nos determina cierta anticipación sobre la respuesta del sistema.

Funcionamiento:

Para lograr que el controlador PID¹¹ regule correctamente la estabilización en 0° los movimientos de alabeo y cabeceo se requiere lo siguiente:

- Un sensor que determine el estado del sistema (ángulo de inclinación de roll y pitch ej. acelerómetro MMA7360).
- Un dispositivo que genere la señal PWM para controlar los servomotores (PIC18F2550).
- Un actuador que modifique al sistema (servomotores).

Basándonos en el error calculado entre el valor actual del ángulo de inclinación y el que queremos alcanzar (0° respecto a la horizontal) enviamos un valor distinto de PWM para lograr que el servomotor gire en uno u otro sentido hasta que el error sea mínimo.

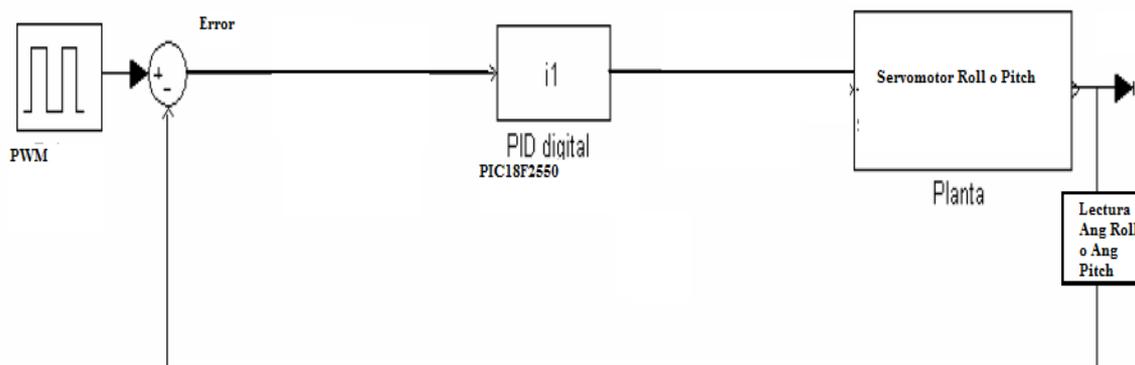


Figura 41. Diagrama de bloques control PID.

El algoritmo a seguir es el siguiente:

- Leemos el error en el ángulo de alabeo y cabeceo (ángulo de inclinación actual).
- Calculamos la salida PWM basándonos en la diferencia de salida y entrada (PID).
- Actualizamos las variables.
- Esperamos un tiempo de muestreo.
- Volvemos a leer el error.

¹¹ Si se requiere saber más acerca de cómo implementar técnicas de control basadas en microcontroladores se recomienda ver [35] y [36].

Una vez contemplados todos los componentes y aspectos anteriores se procedió a la fabricación de la tarjeta de piloto automático.

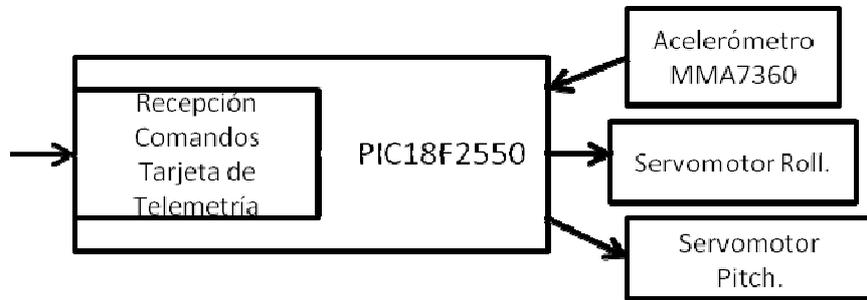


Figura 42. Diagrama de bloques básico para el funcionamiento del piloto automático.

Su aspecto final es el siguiente:

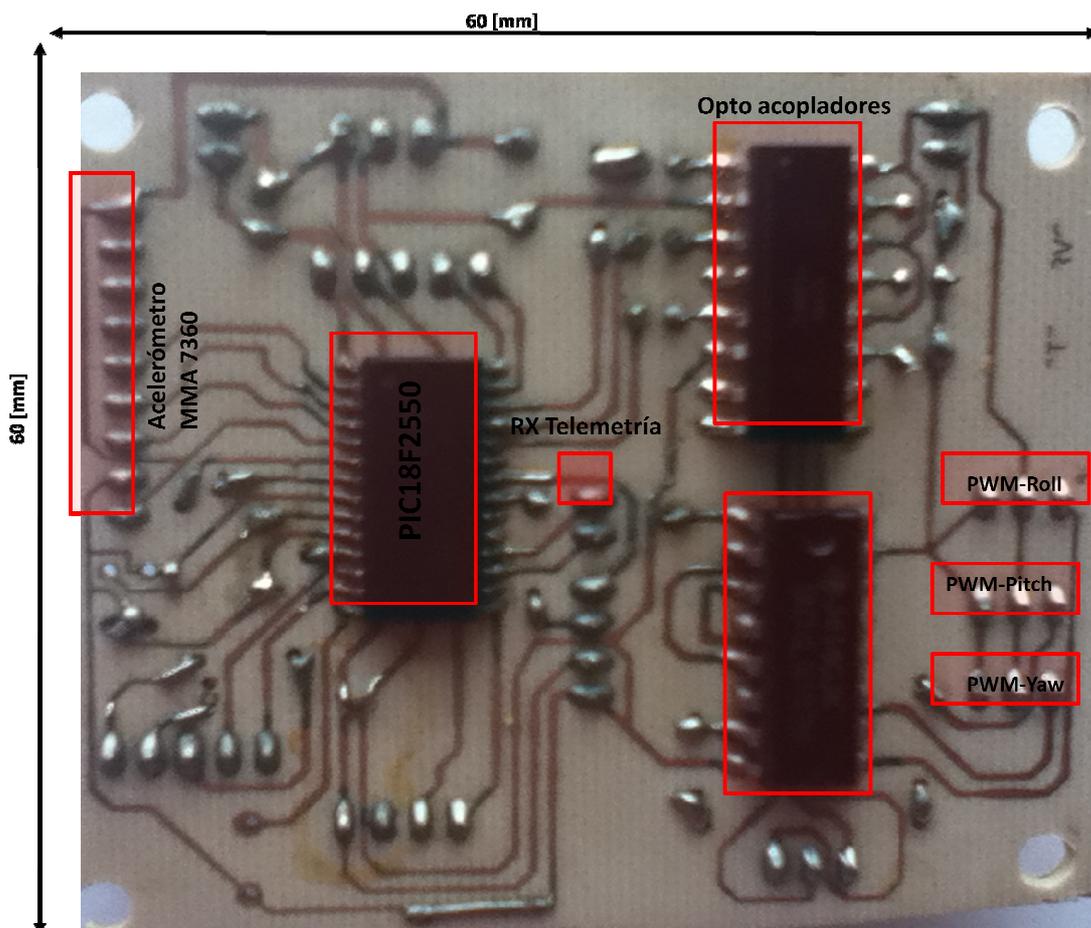


Figura 43. Circuito impreso – piloto automático.

Una vez fabricada la tarjeta de piloto automático se planteó el siguiente diagrama de flujo cuya programación se muestra en el Anexo II.

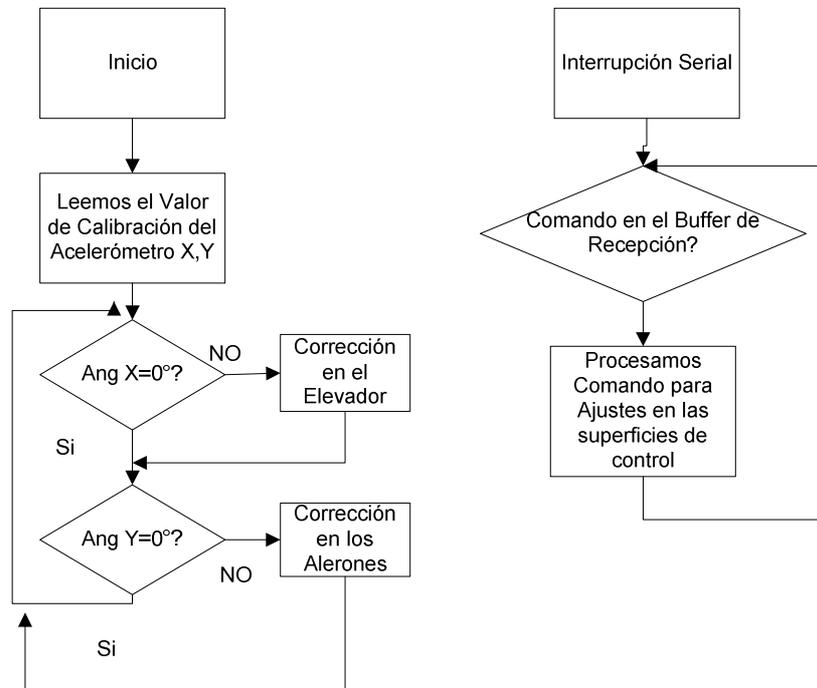


Figura 44. Diagrama de flujo para la programación del piloto automático.

3.6.9 Sistema de seguridad anti fallos (Fail-Safe System).

Con el fin de recuperar el prototipo en caso de contingencia debido a pérdida de comunicación o falla de los algoritmos se ha implementado un subsistema independiente a las tarjetas electrónicas elaboradas, conectado a un canal auxiliar del transmisor de radiocontrol y energizado con la alimentación del mismo(ver Fig. 45).

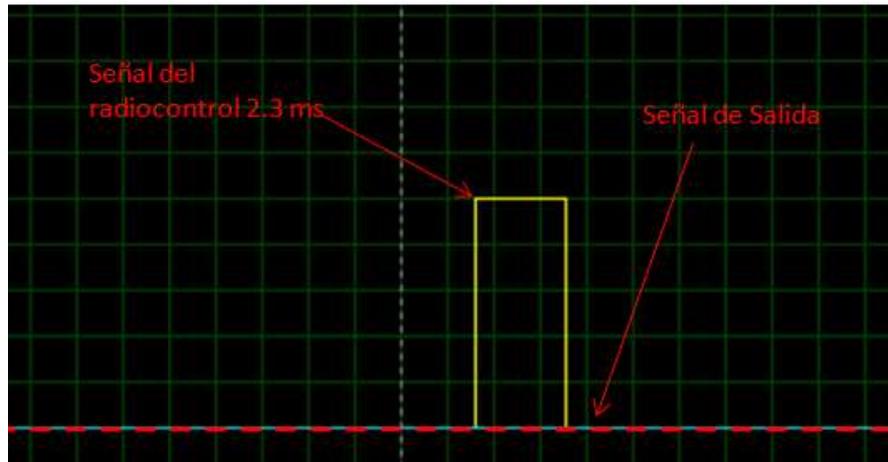


Figura 47.Switch ON- $X=1[V/div]$, $Y=1.2[ms/div]$.

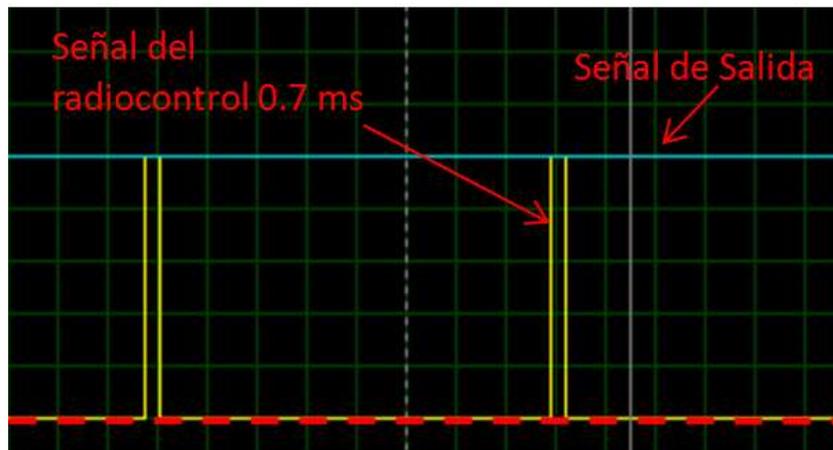


Figura 48.Switch OFF- $X=1[V/div]$, $Y=2.45[ms/div]$.

3.7 Estación de control terrena.

Para mantener en contacto la estación terrena (ver Fig. 50) se requiere de un medio de comunicación inalámbrico el cual permita monitorear los datos enviados por el aeromodelo (meteorológicas, orientación, posición geográfica) así como controlar y verificar el estado de los actuadores del sistema. La comunicación se debe implementar ya sea mediante algún cifrado o ser transparente, teniendo como premisa principal evitar la interferencia entre medios de comunicación (radio control del aeromodelo y sistema de comunicaciones para la estación terrena).

Se propuso el uso de módulos de comunicación inalámbrica que integren receptor y transmisor de comunicación con interfaz serial, pues es uno de los protocolos mas utilizados en los microcontroladores, al ser de este tipo, son fácilmente adaptados al estándar RS-232 en cualquier computadora personal.

Por lo anterior se consideró el módulo de radiofrecuencia HM-TR915 dejando como segunda opción, en caso de no cumplir con alguna necesidad, los módulos comerciales ZigBee¹².

3.7.1 Descripción del módulo de comunicaciones.

El transmisor HM-TR915 es un transmisor/receptor de datos inalámbrico que transmite en la banda del espectro electromagnético UHF en 300 Mhz, el modulo es adecuado ya que trabaja en una frecuencia totalmente diferente al receptor del avión (2.4 GHz).

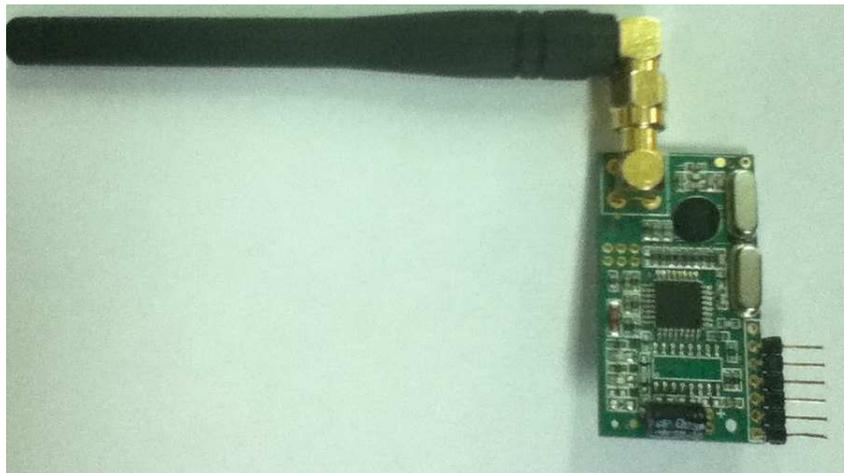


Figura 49. Modulo HM-TR915.

Sus características generales son:

- Alta transmisión de datos.
- Largas distancias de transmisión.
- Frecuencias programables.
- UART configurable.
- Función de bajo consumo (sleep).

¹² Zbee – Pro OEM. RF. (ver www.digi.com).

Entre las características por las cuales fueron escogidos estos módulos destacan las siguientes:

1. Modulación FSK alta inmunidad a la interferencia.
2. Comunicación de 2 vías (half-duplex).
3. Formato de la UART configurable (300-19200 bps).
4. Alto rendimiento en largas distancias de transmisión >300m en campo abierto.
5. Interfaz UART estándar con niveles lógicos TTL u RS232.
6. Consumo máximo de corriente TX @30 [mA] y RX@17[mA].

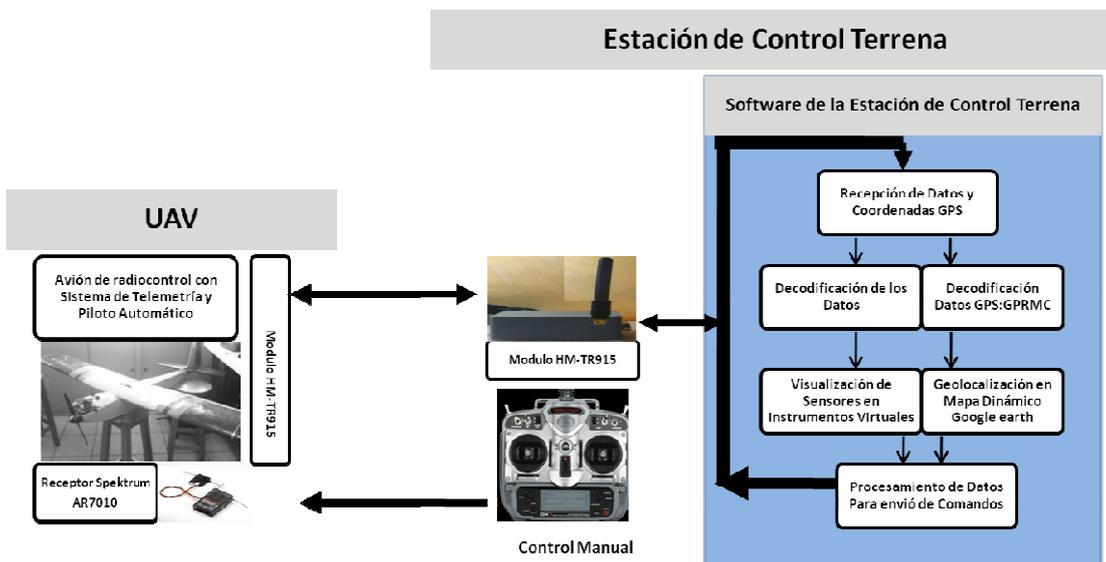


Figura 50. Diagrama de bloques estación de control terrestre.

Para el desarrollo del software de la G.C.S se utilizó el lenguaje de programación Labview de National Instruments, éste es un lenguaje gráfico en el cual es posible implementar funciones matemáticas, despliegado de datos mediante gráficos y mostrar instrumentos virtuales.

3.7.2 Especificaciones del software G.C.S.

Los requerimientos de nuestra interfaz gráfica son los siguientes:

- Enviar y recibir datos a través de una interfaz serial mediante el módulo de comunicaciones HM-TR915.
- Desplegar el estado de las variables en distintos instrumentos virtuales: altímetro, brújula, termómetro etc.
- Mostrar la ubicación del aeromodelo mediante un mapa en coordenadas de latitud y longitud.

3.7.3 Desarrollo del software.

Definidos los puntos de interés en la G.C.S procedemos a mostrar la programación de las partes principales.

Inicialmente se programa la interfaz de comunicaciones, esto se realizó mediante el VI Visa serial (ver Fig. 51), con este es posible inicializar el puerto serial de la computadora así como configurarlo.

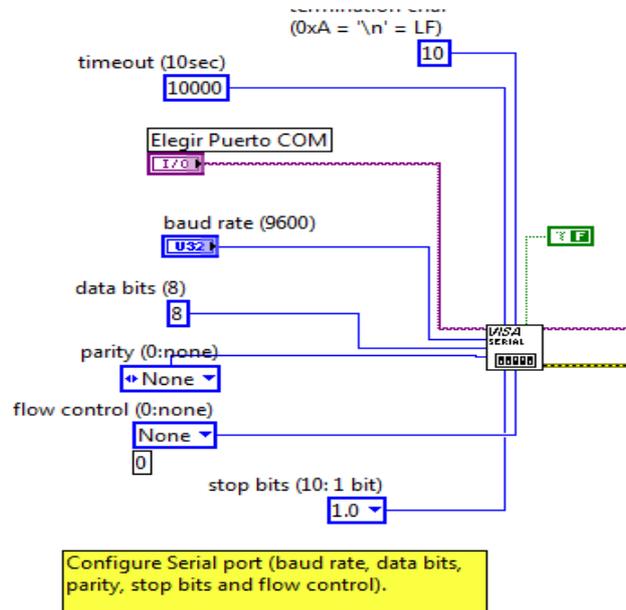


Figura 51. Configuración e inicialización del puerto serial.

Como se definió en la reglas de comunicación los datos son enviados en un formato predefinido, para identificarlos se usa la función *Match Pattern Function*, esta nos permite

buscar caracteres o expresiones específicas y separarlos, como se había mencionado separamos cada dato de la forma * $CX_n Nvalor_n$ para realizar los cálculos que sean necesarios y así desplegar los valores mediante un instrumento virtual (ver Fig. 52).

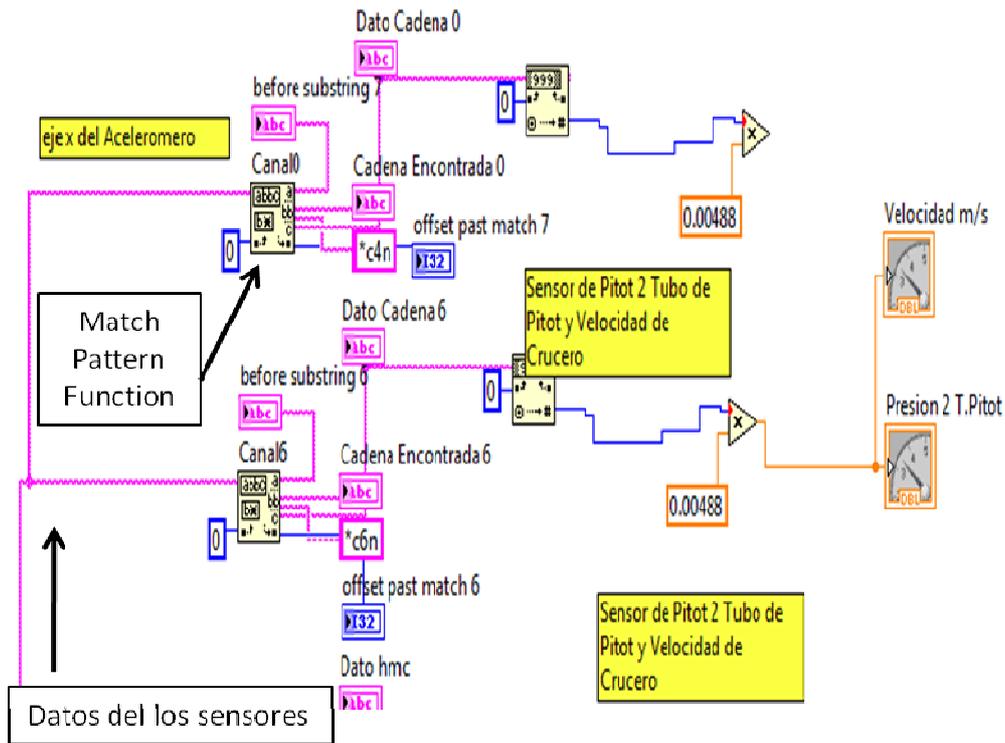


Figura 52. Separación de datos.

Al igual que con los datos de los sensores, realizamos la misma acción para la sentencia GPRMC (ver Fig. 53) proveniente del GPS, una vez hecho esto mediante la API¹³ de Google Earth¹⁴, disponible en la página de National Instruments, se actualiza un archivo kml el cual es leído por este software permitiéndonos ver la ubicación del aeromodelo.

El kml es un lenguaje de marcado basado en XML para representar datos geográficos en tres dimensiones, actualmente es desarrollado por Google inc¹⁵.

¹³ Visitar <https://decibel.ni.com/content/docs/DOC-7983>.

¹⁴ Google Earth es un Software de la empresa Google el cual permite ver imágenes satelitales, mapas o imágenes de relieve del planeta.

¹⁵ Google inc. es una empresa cuyo principal producto es el motor de búsqueda de contenido en Internet del mismo nombre.

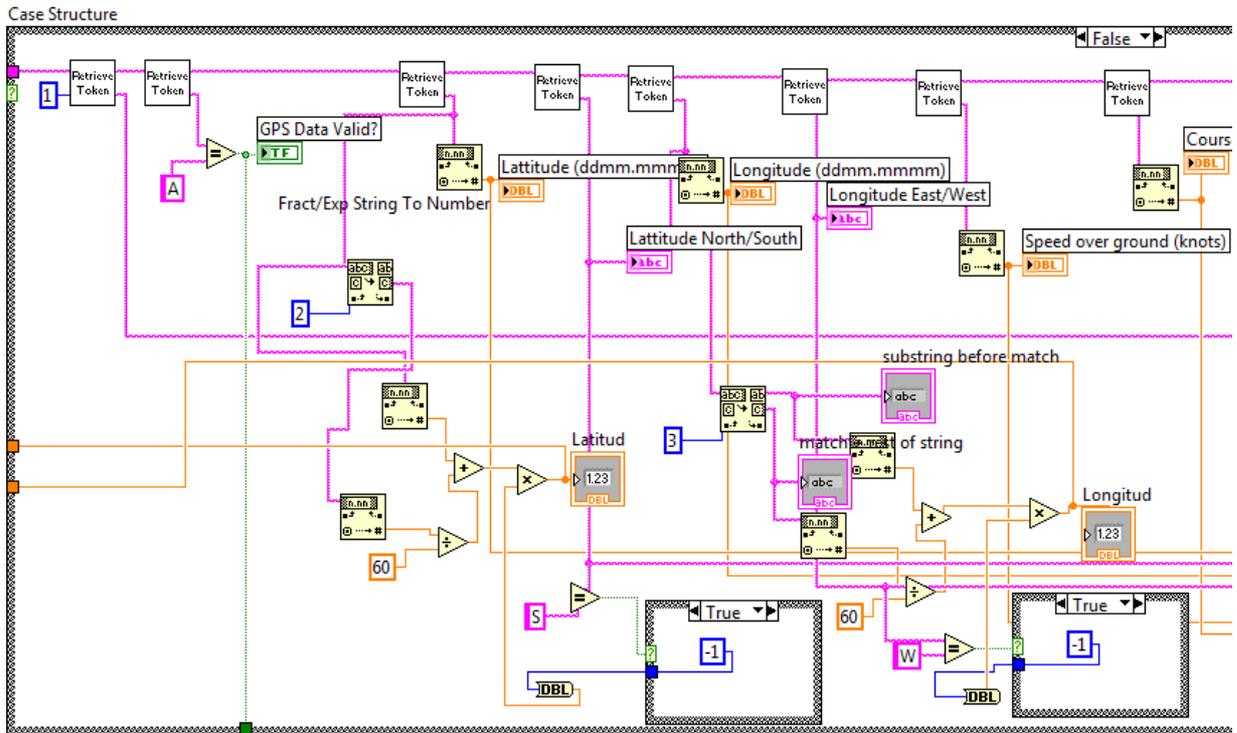


Figura 53. Separación de datos GPS.

Terminada la programación se realizó la interfaz gráfica que permite visualizar en instrumentos virtuales el valor de los sensores así como ubicar el aeromodelo mediante un mapa satelital.



Figura 54. Interfaz gráfica para la estación de control terrena.

IV Realización de pruebas y adquisición de datos.

Para la realización de las pruebas, se contó con la ayuda de un club de aerodelismo local del cual recibí asesoría en todo momento respecto al vuelo de aerodelos de radiocontrol.



Figura 55. Verificación del aerodelo en tierra.

Basándonos en los objetivos planteados del Capítulo I se desarrollaron las siguientes pruebas:

1. Verificación del envío de datos de la tarjeta de telemetría y funcionamiento del piloto automático (ver Fig. 56 y 60).
2. Verificación del F.S.S. (ver Fig. 47 y 48).
3. Recepción de datos en el software de la estación de control terrena (ver Fig. 58).

Los datos recibidos en la estación de control terrena pueden ser visualizados de manera inmediata con la interfaz gráfica o en modo texto con algún visor de comunicación serial y guardados en un archivo de texto para un análisis posterior.

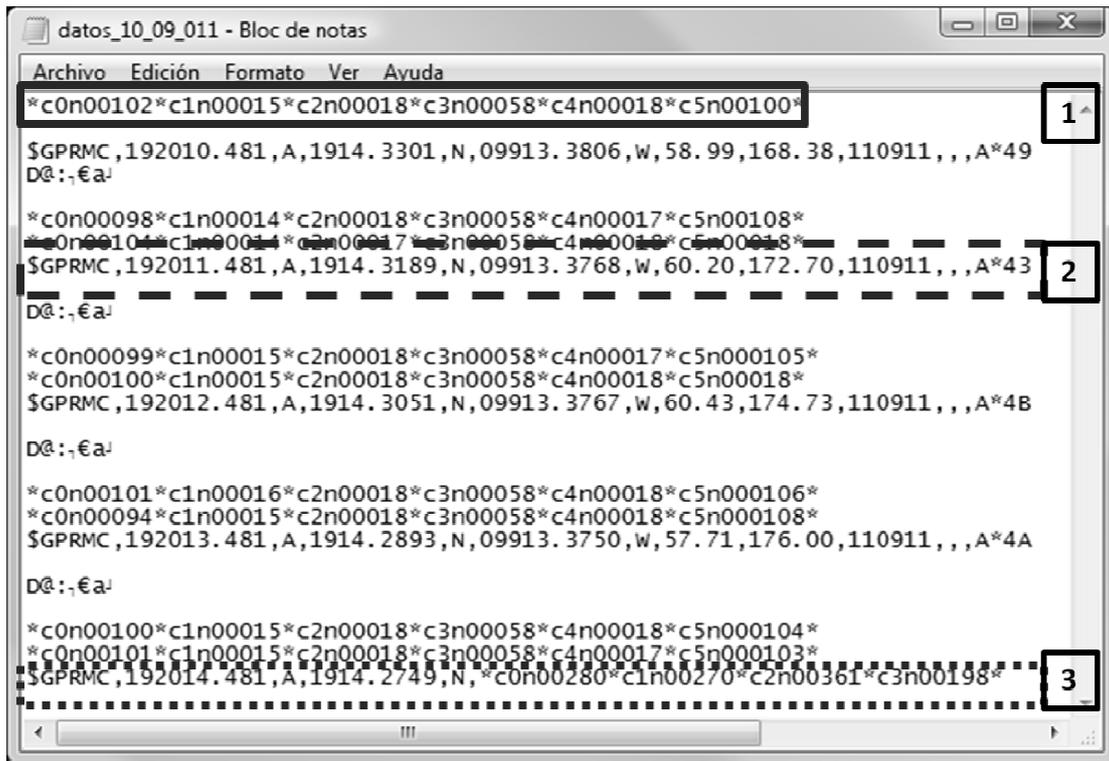


Figura 56. Adquisición de datos: 1. Datos de los sensores, 2. Datos GPS, 3. Sentencia errónea.

En la figura 56 se muestran los datos enviados por el sistema de telemetría, se aprecia enmarcada en la parte superior y con el numero uno la sentencia de datos que contiene el valor de los sensores.

*c0n00102*c1n00015*c2n00018*c3n00058*c4n00018*c5n00100*

Desglosando esta cadena obtenemos la siguiente tabla:

Tabla 6. Desglose de la cadena numero 1 (ver Fig.56).

Variable	Valor	Unidad
Presión	102	[kPA]
Temperatura 1	15	[°C]
Temperatura 2	18	[°C]
Humedad	58	[%RH]
Rapidez	18	[m/s]
Altitud	100	[m]

Con el número 2 se encuentran enmarcados los datos provenientes del GPS cuya cadena es la siguiente:

```
$GPRMC,192011.481,A,1914.3189,N,09913.3768,W,60.20,172.70,110911,,A*43
```

Resaltado en negrita vemos el carácter “A” el cual indica que los datos son validos (ver tabla 3).

Las coordenadas contenidas en esta sentencia son:

Latitud = 1914.3198, N Longitud = 09913.3678, W

Para visualizar estas coordenadas se requiere convertirlas al formato estándar de Google Maps, usando la herramienta en línea GPS Visualizer¹⁶ vemos que el punto analizado es correcto ya que nos muestra la pista de vuelo (ver Fig. 57).



Figura 57. Mapa creado con GPS-Visualizer para las coordenadas: Lat = 1914.3198, N y Lon = 09913.3678, W.

¹⁶ Ver www.gpsvisualizer.com

Por último, con el numero 3 se encuentra enmarcada una sentencia errónea, la cual es descartada automáticamente por el software.

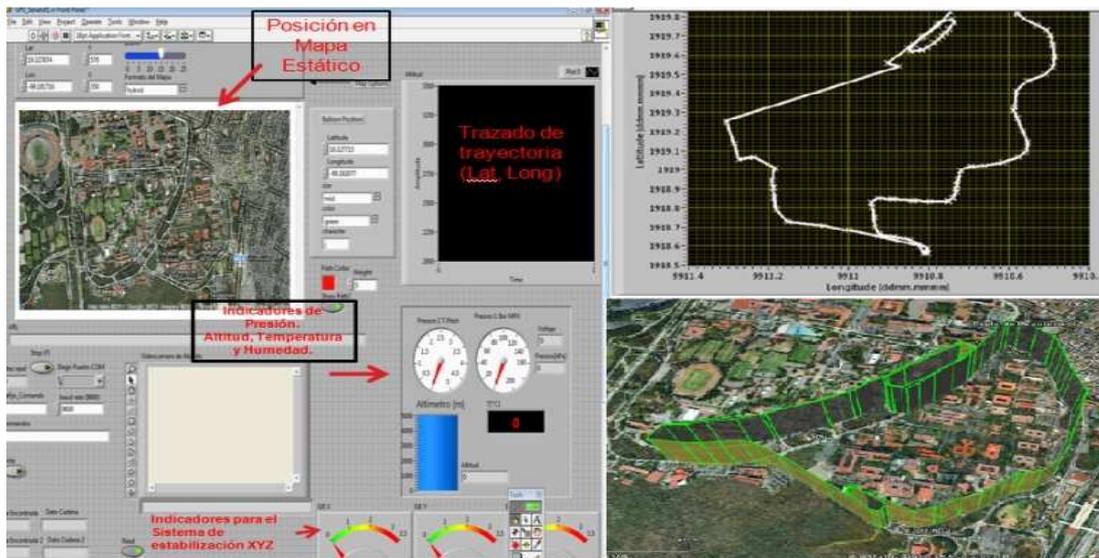


Figura 58. Interfaz gráfica para la estación de control terrena mostrando el trazado de una ruta dentro del circuito de ciudad universitaria.

Para probar el funcionamiento de la interfaz gráfica, se recolectaron datos del sistema de telemetría para trazar una ruta aleatoria dentro de la ciudad universitaria dando como resultado la figura anterior.

V Resultados de las pruebas

Se verificó el funcionamiento del sistema Aeromodelo - Estación de control terrena adquiriendo los datos remotamente y visualizándolos con el uso de la interfaz gráfica.

En la parte inferior derecha de la figura 59 se muestran los datos del aeromodelo: se aprecia la presión barométrica del lugar de pruebas así como la velocidad del aeromodelo, su altitud, temperatura exterior e interior, humedad del ambiente y el rumbo que sigue en un determinado instante de tiempo.

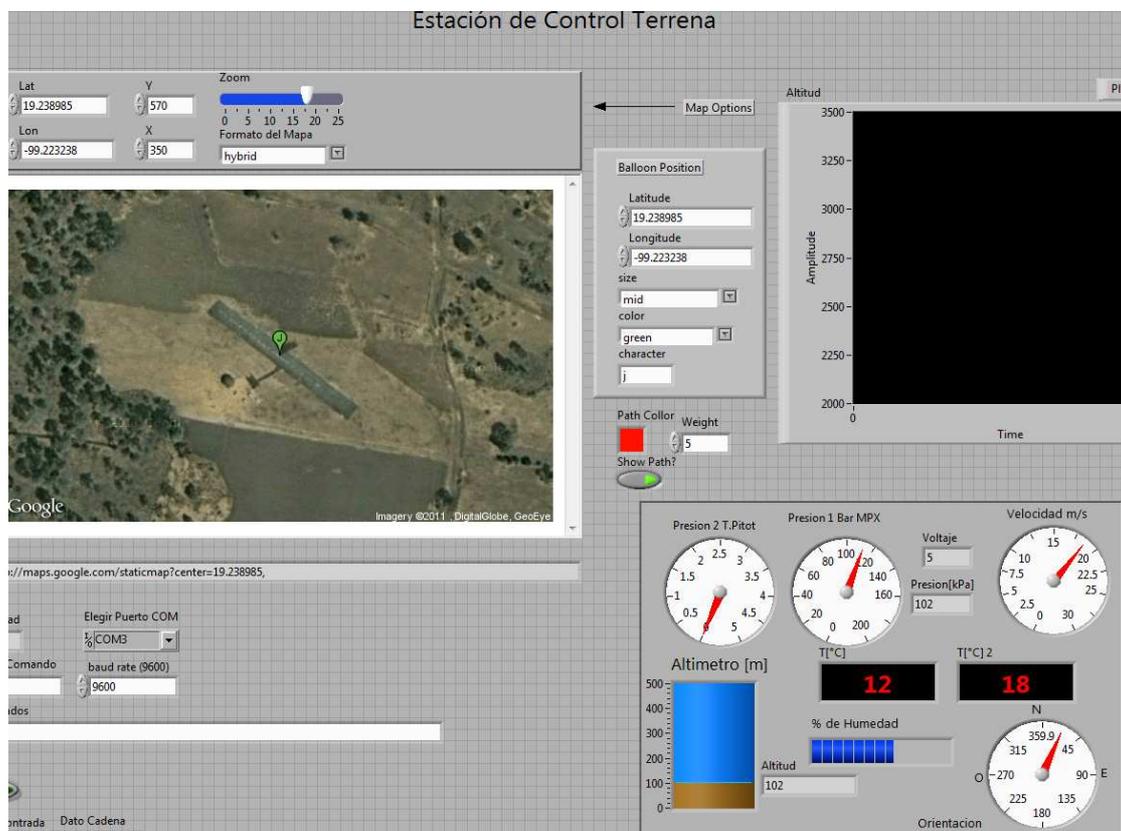


Figura 59. Visualización de datos en la interfaz gráfica.

En la figura 60 y 61 se aprecia el trazado de la ruta seguido por el avión en pleno vuelo, en los puntos que están unidos por líneas se desactivo el piloto automático, por ejemplo del punto 8 al 9 se utilizó el control manual y del 9 al 10 entró en funcionamiento el sistema durante 3 segundos, este no permaneció encendido en todo momento debido a que se vio afectado por la vibración que produce el motor.



Figura 60. Vista 1: Trazado de la ruta y puntos de geolocalización del aeromodelo



Figura 61. Vista 2: Trazado de la ruta y puntos de geolocalización del aeromodelo.

Finalmente en la figura 62 se muestra el alcance máximo registrado en el sistema de comunicaciones, el cual fue de 168 m.



Figura 62. Rango máximo de alcance registrado mediante el modulo de comunicaciones HM-TR915.

Los resultados anteriores son satisfactorios ya que se logró monitorear la ubicación del aeromodelo en un mapa satelital y visualizar los sensores en la interfaz gráfica, en cuanto al piloto automático se obtuvo un desempeño del 30%.

VI Conclusiones

Se han cumplido los objetivos planteados al principio de este trabajo:

- Se diseñaron, fabricaron e integraron los instrumentos de vuelo necesarios para mejorar el desempeño de un aeromodelo de radiocontrol y emular a un avión no tripulado, esto es: se instrumentó electrónicamente el aeromodelo para monitorear diversos parámetros, los cuales son visualizados a través de una interfaz gráfica.
- Mediante la implementación de un F.M.S. y la integración de los sensores así como programación de algoritmos en el procesamiento de datos, decodificación de parámetros para geolocalización y respuesta automática de los actuadores, se realizó un piloto automático capaz de controlar los movimientos de alabeo y cabeceo de manera autónoma en un 30%.
- Para completar la autonomía del aeromodelo, se ha desarrollado mediante el software Labview, una interfaz gráfica que completa la Estación de Control Terrena con el cual se monitorean los sensores, a su vez este nos permite visualizar la ubicación geográfica del aeromodelo en un mapa satelital.

6.1 Trabajo Futuro

Ante la inminente utilización de tecnologías UAV para estudios atmosféricos, vigilancia, o aplicaciones civiles y militares, el trabajo desarrollado en esta tesis da la pauta para hacer mejoras en cuanto a la autonomía del aparato así como velocidad, alcance, techo y mejoras en la instrumentación.

Cada una de las partes desarrolladas podrían ser mejoradas notablemente usando otras tecnologías o dispositivos: microcontroladores más veloces y con mayor memoria, sensores más modernos, sistemas de comunicación mas robustos etc.

Un aspecto importante seria agregar redundancia a los sistemas de control, comunicaciones, electrónicos y de suministro de energía; con lo cual se mejoraría el rendimiento del avión y sobre todo asegurar su funcionamiento en caso de alguna falla.

En cuanto al software para la estación de control terrena, este se desarrolló en LabView por rapidez y disponibilidad de soporte, sin embargo, también es posible generar interfaces de usuario en otros entornos de programación (Ej. Visual Basic, C#, Java) o sistemas operativos distintos (Ej. Linux, MAC OS) buscando el mejor rendimiento y robustez al finalizar la aplicación.

Bibliografía.

- [1] Kimon P. Valvanis. **Advances in Unmanned Aerial Vehicles**. Springer, págs. 3-42:49-72, 2007.
- [2] David G.Hull .**Fundamentals of Airplane Flight Mechanics**. Springer, págs. 16-32:237-239, 2007.
- [3] John H.Blakelock. **Aircraft and Missiles**- Wiley. Jonhn Wiley Inc, 1965.
- [4] Clifford Matthews. **Aeronautical engineer's Data Book** .ButterWorth Heinemann, págs. 66-75,106-114, 2002.
- [5] George M. Siouris. **Aerospace Avionics Systems A Modern Synthesis**. Academic Press Inc, págs. 406-441, 1993.
- [6] Moir And Allan Seabridge. **Aircraft systems 3ra Ed** . Wiley, págs. 443-460, 2008 .
- [7] Ramjee Prasad. **Applied Satellite Navigation**. Artech House, págs. 39-56, 2005.
- [8] Abusleme Hoffman. **Tesis Control Difuso de un Vehículo Volador No Tripulado**. Pontificia Universidad Católica de Chile, págs. 125-148 , 2000.
- [9] Luque Vega F . **Tesis Diseño, Construcción y Control de un Helicóptero de 4 Rotores**. CINVESTAV UG, 2010.
- [10] Michael V.Cook . **Flight Dynamics Principles**. Elseiver, págs. 32-64:66-96, 2007.
- [11] Roger W.Pratt. **Flight Control Systems, Practical Issues in Design and Implementation**. The Institution of electrical engineers , págs. 170-173, 2000.
- [12] James Bao ,Yen Tsui. **Fundamentals of Global Positioning Systems Receivers**.Wiley,paginas , págs. 2-5, 2005.
- [13] Mohinder. **Global Positioning systems, Inertial Navigation, and Integration**. Wiley, 2007.
- [14] Esmat Bekir. **Introduction to Modern Navigation Systems**. Wolrd Scientific, págs. 1-6, 2007.
- [15] Martin Simons. **Model Aircraft Aerodynamics**. Argus Books, págs. 83-100, 1994.
- [16] Carmona Isidoro . **Aerodinámica y Actuaciones del Avión**. Paraninfo, págs. 49-57:358-390:400, 2000.

- [17] John J.Bertin, Michael L.Smith. **Aerodynamics for Engineers**. Prentice Hall, págs. 37-42:40:81-85, 1979.
- [18] Jesús Martínez Rueda. **Sistemas eléctricos y electrónicos de las aeronaves**. Paraninfo, págs. 285-463, 2007.
- [19] Ira H.Abbott. **Theory of Wing Sections**. Dover, 1959.
- [20] Tipos de estabilizadores y Alas.
Ver: <http://virtualskies.arc.nasa.gov/aeronautics/4.html>
- [21] Reg Austin. **Unmanned Aircraft Systems, UAVS Design, Development and Deployment**. Wiley, págs. 1-15:169:183-195, 2010.
- [22] Vergara Diaz, Aguledo Toro. **Diseño e implementación del sistema de Comunicaciones Basado en CAN para la aviónica de un Vehículo Aéreo No tripulado**. Revista Ingenierías Universidad de Medellín-Colombia, 2007.
- [23] Cary R.Spitzer. **The Avionics Handbook**. CRC Press, págs. 13-16, 2001.
- [24] Bill Yenne. **Attack of the drones , A History of Unmanned Aerial Combat**.Zenith Press, págs. 11-18, 2004.
- [25] Generalidades en aviones de R/C . Ver: <http://www.rc-airplane-world.com/>
- [26] Microchip. Hoja de datos **PIC18F4550/2550**. Microchip Technology Inc.
- [27] Microchip. Hoja de datos Convertidor A/D 16 Bits **MCP3425**. Microchip Technology Inc.
- [28] HoneyWell. Hoja de Datos Compas Digital **HMC6352**.
- [29] Ag electrónica. Hoja de Datos Sensor de Humedad. **HMC-433A1**.
- [30] Manual : **The I2C bus specification Version 2.1 January 2000**. Philips Semiconductors.
- [31] Microchip. Hoja de datos Sensor de temperatura MCP9800. Microchip Technology Inc.
- [32] Freescale Semiconductor. Hoja de Datos Sensor de Presión **MPXAZ6115A**. 2007.
- [33] Freescale Semiconductor. Hoja de Datos Sensor de Presion **MPX2010**. 2007.
- [34] Freescale Semiconductor. Nota de Aplicación: **Understanding Pressure and**

Pressure Measurement. 2005.

- [35] Linxtechnologies. Hoja de Datos **GPS RXM-GPS-SR.** 2010.
- [36] D. Ibrahim . **Microcontroller based Applied Digital Control.** John Wiley & Sons, 2006.
- [37] D. Ibrahim . **Microcontroller based temperatura Monitoring and Control.** Elsevier Science & Technology Books, 2002.

Apéndice I

I Breve historia de la aviación y perspectiva de los UAV.

1.1 Primeros diseños de UAV

Existe documentación de que la mayor contribución a los mecanismos autónomos ocurrió durante la era de Pitágoras, quien fuera estudiante de Tales algunos años.

El progreso de los mecanismos autónomos se le atribuye a Archytas de Tarantine, de igual forma a Leonardo Da Vinci, el diseño en 1483 un aeronave capaz de rotar, llamada giroscopio u tornillo aéreo (Ver Fig. 1).

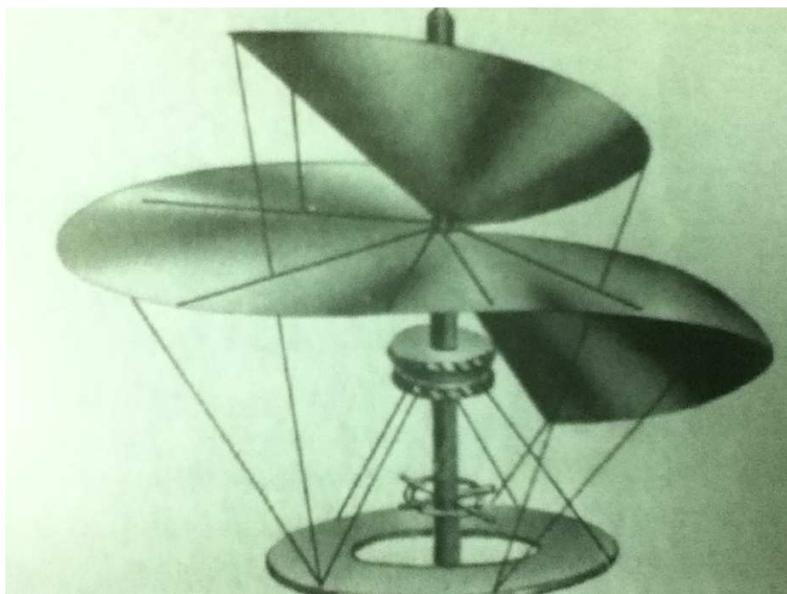


Figura 63. Tornillo aéreo – Leonardo Da Vinci.

Dicha maquina es considerara como el antecesor de lo que hoy conocemos como helicóptero.

Posteriormente, en 1840, se diseñó una máquina voladora vertical por Horacio Phillips, esta se componía por un calentador que se usaba para generar vapor el cual era arrojado hacia las aspas de la hélice [1].

1.2 Actualidades en los UAVs.

Hoy en día existen una gran diversidad de configuraciones para los UAV, éstas incluyen gran escala, mediana y miniUAVs. Actualmente, varios modelos son utilizados por la industria militar alrededor del mundo, una de las principales ventajas sobre los vehículos tripulados es el rendimiento (debido a la ausencia de fatiga en un piloto). A continuación se muestran algunos tipos de UAVs que se encuentran en servicio:

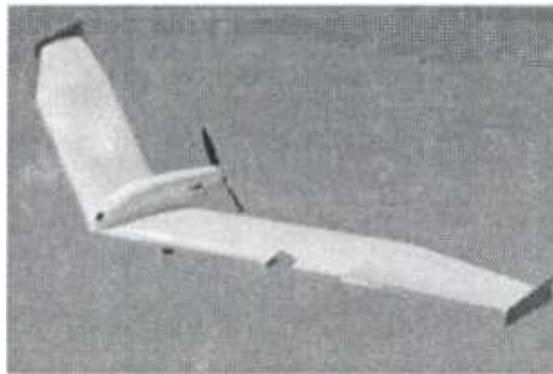


Figura 64.El Seagull construido por Elbitsystems, Israel.

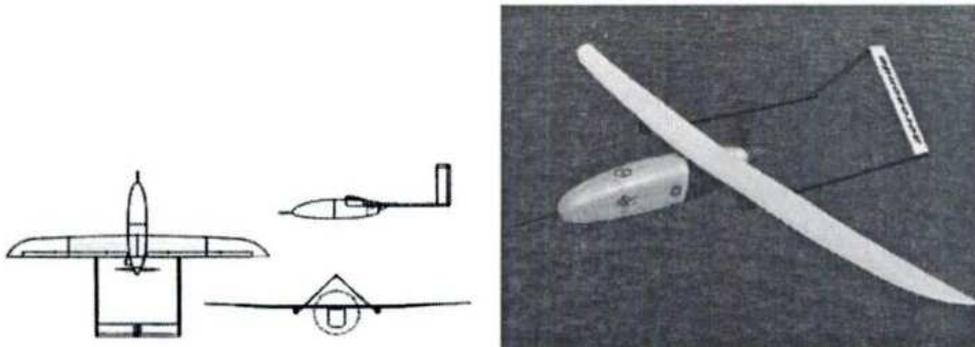


Figura 65.El RQ-2B Pioneer, diseñado por Pioneer UAV Inc. Operado por la marina de los Estados Unidos de América.

En lo que se refiere a UAVs la investigación no solo se limita a la autonomía sino a probar nuevas configuraciones, hoy en día existen modelos no convencionales como el Seagul (ver Fig. 66) de AeroViroment que carece de estabilizador horizontal trasero.



Figura 66.UAV Seagul

Otro modelo interesante es el DP-5X de DragonFly, el cual está diseñado para tareas de reconocimiento, vigilancia, y adquisición de datos del blanco.



Figura 67.El DP-5X de DragonFly

Al igual que los anteriores UAV existen desarrollos para realizar microUAVs buscando reducir su tamaño y aumento en el rendimiento energético.



Figura 68. Vehículo aéreo de reconocimiento de objetivos con microcámara.



Figura 69. Vehículo aéreo comparado con la escala de un lápiz.

Las figuras anteriores representan solo una parte del actual estado del arte en los diseños y aplicaciones de los UAVs.

Como se observa, la mayoría de este tipo de aeronaves han sido desarrolladas por la industria militar, debido a esto, sus principales aplicaciones son de vigilancia, monitoreo y en algunos casos la entrega de municiones.

Sin embargo, con el desarrollo e investigación en las universidades y centros de investigación se planea buscar aplicaciones civiles, haciendo uso de esta tecnología a un bajo costo [1][21].

II Principios básicos de vuelo.

Con el objetivo de conocer estos principios es necesario conocer las fuerzas que se involucran en su movimiento y su origen. A su vez debemos conocer las superficies de control y entender la forma en que estas guían al avión a través del espacio.

Para la explicación de estos principios se pondrá como ejemplo una avioneta modelo Cessna 172 (ver Fig .70).



Figura 70. Avioneta Cessna 172

Este avión monomotor de ala alta es un modelo estable en vientos de baja velocidad, es utilizado comúnmente para operaciones de búsqueda y rescate.

Los aviones se mueven libremente en el espacio de 3 dimensiones, estos tienen 6 grados de libertad, 3 grados corresponden al desplazamiento y los otros 3 a giros en torno a sus ejes.

El movimiento rotacional está relacionado con el movimiento de la aeronave alrededor de los tres ejes, dichos giros tienen nombres específicos y representan maniobras comunes, su nombre está relacionado en torno al eje que se producen: roll , pitch y yaw [2][3][8]. (ver Fig.71).

- **Roll** (Balanceo): Giro en torno al eje X ó eje longitudinal del avión.
- **Pitch** (Cabeceo): Giro en torno al eje Y del avión, la nariz del avión sube o baja.
- **Yaw** (Guiñada): Giro en torno al eje Z del avión, este produce un cambio de dirección horizontal [10].

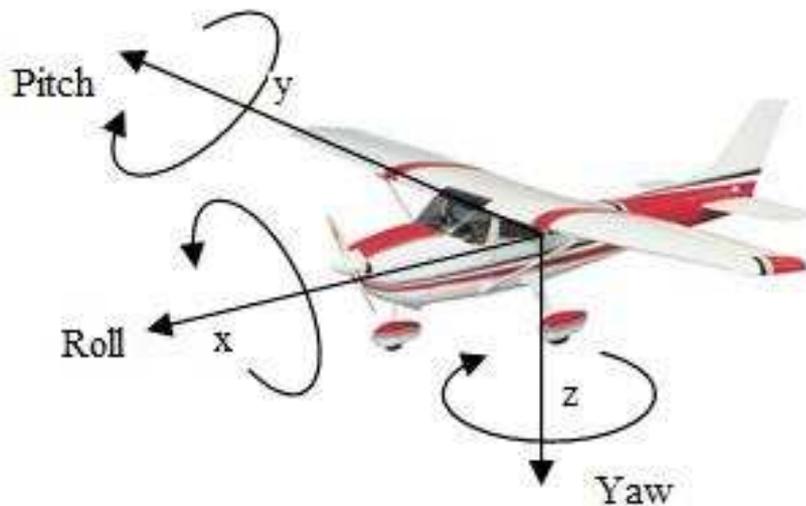


Figura 71. Giros de un avión en torno a sus ejes.

2.1 Sistema de Ejes.

Para desarrollar las ecuaciones de movimiento, es necesario fijar un sistema de ejes ortogonales en el CG (centro de gravedad) del avión, aunque existen diversos sistemas de ejes se utilizará el definido anteriormente.

El eje **OX** longitudinal está situado en el plano de simetría del avión y hacia el morro (frente del avión).

El eje **OY** lateral es el que se encuentra perpendicular al plano de simetría y su sentido positivo es hacia la derecha del ala derecha.

El eje **OZ** vertical situado en el plano de simetría del avión y su sentido positivo es hacia abajo [8][10].

2.2 Fuerzas ejercidas sobre el avión.

Son 4 las fuerzas básicas que producen el movimiento del avión (ver Fig. 72), estas provocan: el despegue, aterrizaje, el ascenso y descenso, así como el vuelo de crucero.

1.- **Lift** (Sustentación.): Fuerza aerodinámica provocada por las alas, esta puede variar dependiendo del perfil.

2.- **Weight**(Peso): Esta fuerza es debido a la fuerza de gravedad y se mantiene constante.

3.- **Thrust**(Empuje): Es la fuerza que produce el motor del avión(motor de combustión, turbina, etc), cuando esta fuerza es igual al arrastre se mantiene la rapidez.

4.- **Drag** (Arrastre): Esta fuerza aerodinámica se opone al movimiento del avión y debe ser pequeña. El arrastre se puede ver incrementado o decrementado dependiendo del tipo de ala, está fuerza se divide en la presión de arrastre y la superficie de fricción de arrastre.

- Presión de arrastre: Está asociado a la formación de una estela detrás del objeto, que genera un "pozo de presión", estas fuerzas aparecen en todos los cuerpos, pero su importancia relativa puede cambiar dependiendo de la forma del mismo.
- Superficie de fricción de arrastre: Fuerza de fricción entre una superficie y un fluido.



Figura 72. Fuerzas ejercidas sobre un avión.

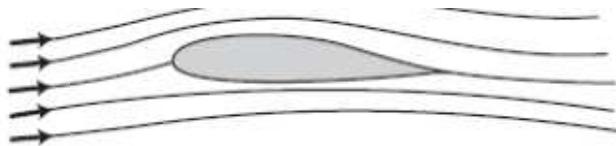


Figura 73. Flujo de aire en un perfil alar.

Las fuerzas anteriores son básicas para el vuelo del avión, la sustentación permite que el avión se eleve contrarrestando el peso debido a la gravedad, produciéndose por diferencia de presiones en las caras del ala(ver Fig. 73), debido a esta diferencia se genera la fuerza necesaria para elevar la aeronave [2][16].

Un aspecto importante a mencionar es que la velocidad del avión se verá afectada dependiendo del ángulo de ataque en el que las alas encaren al viento, esto debido a que el flujo a través de estas será diferente (ver Fig. 75 y 83).

El ángulo de ataque en las alas es aquel que existe entre la cuerda del ala y la dirección de la corriente libre del aire (Ver Fig.12).

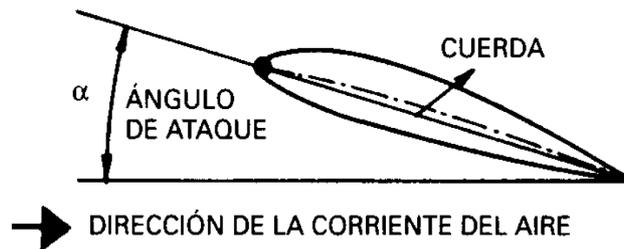


Figura 74. Angulo de ataque.

El ángulo de ataque se encuentra ligado a la velocidad que puede llevar el avión debido a que existe una relación entre el peso y la fuerza de sustentación.



Figura 75. Relación entre ángulo de ataque y velocidad.

Aunque las fuerzas anteriores son las elementales para entender el desplazamiento y movimiento de un avión existen otros factores secundarios que pueden influir como:

- Forma del perfil.
- Superficie de las alas y forma de la superficie.
- Densidad del aire.
- Velocidad.
- Ángulo de ataque.

Si queremos ser más rigurosos a los factores anteriores pueden añadirse otros efectos si se desea realizar un estudio más profundo, agregando la viscosidad y compresibilidad se desprenden otras variables que intervienen en el vuelo como [15][16][19]:

- Velocidad del sonido en el aire
- Viscosidad del aire.
- Rugosidad de la superficie.

2.3 Equilibrio y compensación fina (trimming) de un avión.

Para que la aeronave este en equilibrio se debe mantener el vuelo de crucero y el peso de una manera estable, la sustentación debe ser igual al peso y el empuje que proporciona el motor debe ser igual a la resistencia del viento (arrastre).

En un vuelo normal es común que el piloto ajuste las superficies de control para que si este libera los controles la aeronave se quede en una condición de vuelo preestablecida, si no se realizara esto el piloto tendría que ajustar constantemente las superficies de control, la acción anterior se denomina compensación fina.

A grandes rasgos el objetivo principal al compensar (ver Fig.76) un avión es llevar las fuerzas y momentos que se ejercen a un estado de equilibrio [10].

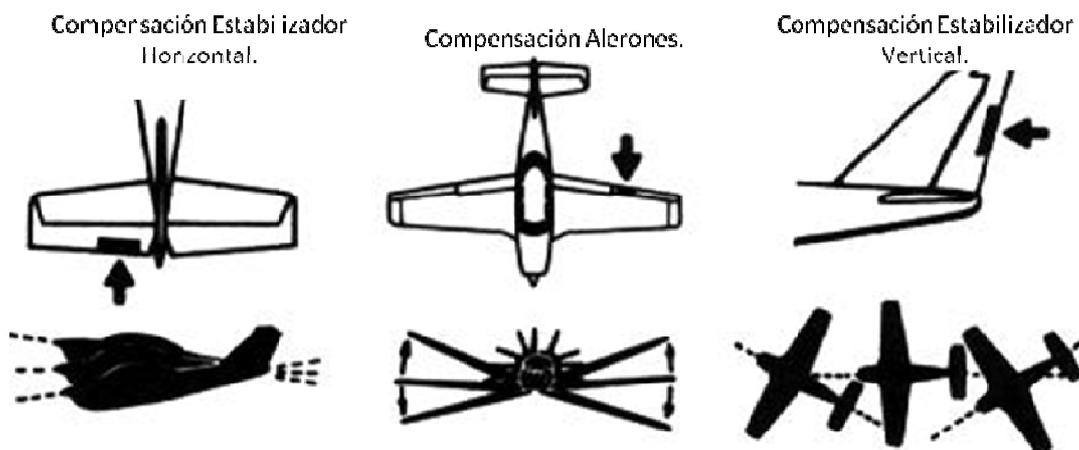


Figura 76. Acciones y movimientos al efectuar las acciones de compensación.

2.4 Estabilidad estática y dinámica.

Si definimos primeramente la estabilidad estática de un cuerpo sabemos que esta es la tendencia inicial para volver a una determinada posición de equilibrio, después de haber experimentado una perturbación que le haya alejado de dicha posición.

Haciendo referencia al avión tenemos que: La estabilidad estática concierne a las acciones de control requeridas para establecer el equilibrio con ciertas características para asegurar que permanezca en ese estado, las características de la estabilidad estática alrededor de los tres ejes determinan la compensación de la aeronave.

Estabilidad dinámica:

Por definición la estabilidad dinámica trata la forma en que un cuerpo se mueve conforme transcurre el tiempo, después de haber sido sometido a una perturbación.

Refiriéndonos al avión, nos determina las características de movimiento transitorias después de una perturbación y alrededor de una condición de vuelo.

Al realizar los ajustes de compensación en una aeronave esta permanecerá en equilibrio, a menos que sea perturbada, debido a que el piloto ajuste las superficies de control ó a influencias externas, por ejemplo, una turbulencia.

La mayoría de estos parámetros tienden a cambiar durante un vuelo por lo que según sea necesario se debe compensar la aeronave, esto no representa una tarea difícil debido a que la simetría de un fuselaje típico nos otorga propiedades simétricas aerodinámicas, que usualmente reducen la tarea, enfocándonos únicamente a realizar el ajuste de compensación de manera longitudinal.

En cuanto a los ajustes laterales-direccionales estos únicamente son requeridos cuando se ha perdido la simetría aerodinámica, por ejemplo en un avión multimotor, al fallar uno de sus motores (ver Fig. 77) se tiene que reajustar las superficies para que este permanezca en equilibrio [16].

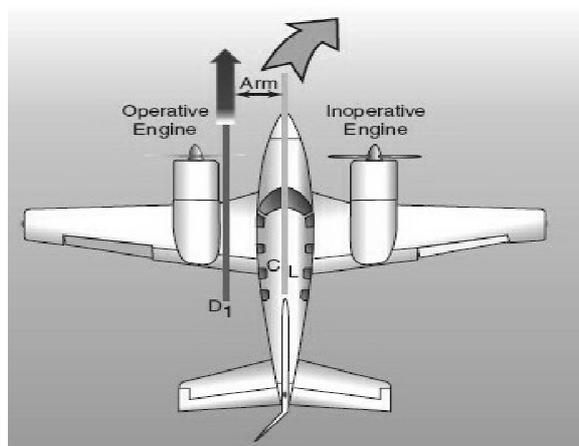


Figura 77. Desequilibrio debido a la pérdida de un motor.

III Superficies de control de una aeronave y estructura.

El desplazamiento de un avión es controlado mediante la manipulación del ángulo de las diversas superficies de control, éstas superficies están sujetas a la estructura del mismo, como se aprecia en la siguiente figura.

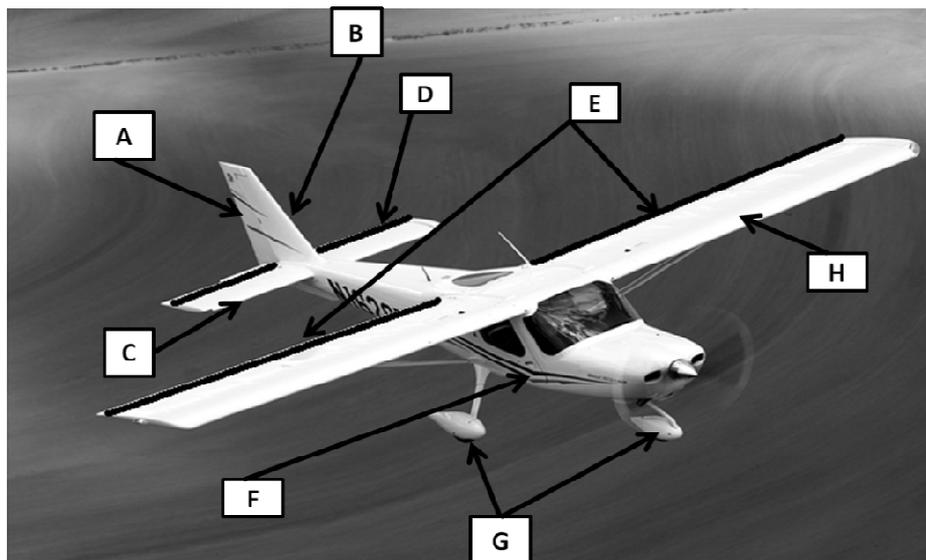


Figura 78. Superficies de control de un avión: A.- timón de la dirección, B.- estabilizador vertical, C.- estabilizador horizontal, D.- elevador, E.- alerones, F.- fuselaje, G.- tren de aterrizaje, H.- alas.

Las acciones de control manipulan los momentos alrededor del centro de gravedad y una acción secundaria es el cambio de las fuerzas aerodinámicas externas.

Las superficies de control y sus acciones son las siguientes:

Estabilizador horizontal: La parte trasera del avión, conocida como estabilizador horizontal, está compuesta por: una parte delantera la cual es fija y un parte móvil llamada elevador. El ángulo de este elevador afecta el movimiento longitudinal, así como el equilibrio de la fuerza de sustentación, al igual que el momento de cabeceo alrededor del centro de gravedad.

Estabilizador vertical: Esta parte trasera del avión nos controla el ángulo de guiñada mediante una parte móvil conocida como timón de la dirección. Por lo general el estabilizador horizontal y vertical se encuentran juntos. El efecto de manipular el timón se

ve reflejado en el cambio de la dirección lateral de movimiento así como en la maniobrabilidad.

Alerones: Estas superficies se encuentran en la parte posterior de las alas del avión, al cambiar el ángulo de estas superficies (ver Fig. 79) podemos controlar el balanceo del mismo, hay que destacar que el ángulo de los alerones tienen direcciones opuestas al moverse y este no necesariamente debe ser el mismo.

Alas: Son la principal fuente de sustentación del avión, estas alojan a los alerones.

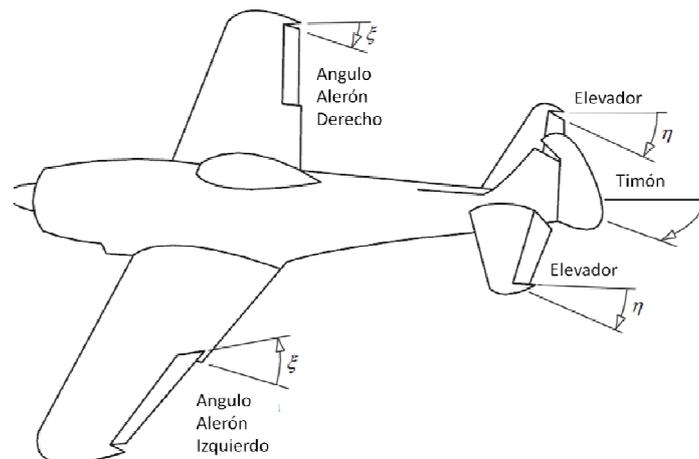


Figura 79. Ángulos en las superficies de control.

IV Estructura del avión.

La estructura de un avión está dividida en las siguientes partes (ver Fig. 78):

Fuselaje: El fuselaje o cuerpo del avión es la estructura en donde se montan las alas, así como el motor y el tren de aterrizaje, esta también puede incluir una cabina para el o los pilotos y pasajeros.

Tren de aterrizaje: Esta parte del avión es la primera en hacer contacto con el suelo en el aterrizaje, dicha sección está calculada para soportar las fuerzas ejercidas al aterrizar, así como para maniobrar en tierra.

Motor: Es la fuerza principal del avión, su función es generar el empuje necesario para que el avión alcance una determinada rapidez y así permitir la interacción de las fuerzas aerodinámicas [11].

V Etapas de vuelo.

Despegue: En esta etapa el avión debe adquirir la suficiente velocidad en la pista para poder generar la suficiente fuerza de sustentación como para vencer a la fuerza de gravedad.

Ascenso: Mediante esta etapa el avión aumenta la altitud de una manera constante, sin una pérdida considerable de velocidad hasta llegar a la altura de seguridad comercial. Por lo general se buscan altitudes mayores donde el aire es menos denso provocando una resistencia menor al avance del avión.

Velocidad de crucero: En esta etapa el avión busca desplazarse a una rapidez y altitud constante, consiguiendo con esto que las fuerzas aerodinámicas se encuentren en equilibrio, dicha etapa se puede realizar a altas velocidades o bajas, cambiando con esto el ángulo de ataque.

Viraje: Etapa de vuelo en la que el avión cambia el rumbo, esta se realiza mediante el control del timón de la dirección, debido a que esta acción de control puede ser muy lenta se usan los alerones provocando la inclinación del avión hacia la dirección que se desea virar, cuando esto sucede el avión tiende a picarse por lo cual se tiende a elevar el timón de profundidad para compensar el ángulo de ataque [16].

Descenso y Aterrizaje: Mediante el descenso el avión se prepara para bajar a la pista de aterrizaje, disminuyendo gradualmente la potencia del motor y por consiguiente su altitud, en esta etapa el avión desciende hasta tocar tierra, para realizar esta última maniobra, el avión debe ir a una rapidez relativamente baja así como elevar la nariz (cambio del ángulo de ataque, ver Fig.80) para no forzar el tren de aterrizaje.

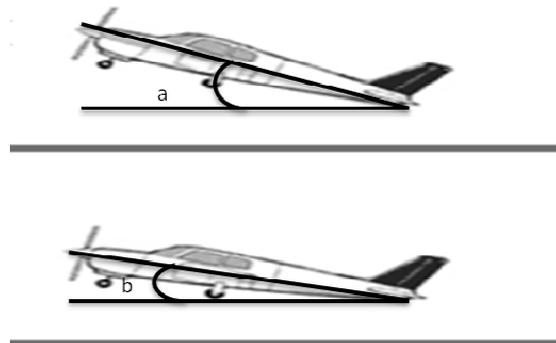


Figura 80. Cambio del ángulo de ataque en el aterrizaje.

VI Ecuaciones básicas de movimiento de un aeroplano.

Introducción:

En este capítulo describimos las ecuaciones básicas de movimiento de un aeroplano en vuelo, describiendo sus propiedades cinemáticas y ecuaciones generales[2][3][8][17].

Las ecuaciones de movimiento son derivadas de la segunda ley de Newton como una suma de vectores que incluye las fuerzas y momentos aplicados al aeroplano relativo a un marco de referencia inercial.

Ecuaciones de movimiento:

El movimiento de un avión, como el de cualquier otro cuerpo se estudia por medio de la segunda ley de Newton, esta nos dice que la resultante de todas las fuerzas que actúan sobre él es igual a:

$$\sum F = Masa \times Aceleración Lineal \quad (1)$$

Para un objeto desplazándose a una velocidad variable tenemos que la suma de fuerzas que interactúan en el es igual a:

$$\sum F = \frac{d}{dt}(mV) \quad (2)$$

Lo anterior nos determina la tasa de cambio de su cantidad de movimiento, en cuanto a la suma de los momentos, ésta se determina mediante:

$$\sum M = \frac{dH}{dt} \quad (3)$$

Lo cual nos da la tasa de cambio del momento angular, todo lo anterior respecto al sistema de referencia inercial.

Describiendo los términos utilizados en las ecuaciones anteriores tenemos que:

F= Fuerza

M= Momento

m= masa del avión.

V= velocidad

H= es el momento angular.

Considerando la masa del avión constante, podemos obtener la expresión para obtener su aceleración, recordando que esta puede cambiar su tamaño y dirección. Para obtener la aceleración tenemos la siguiente expresión:

$$a = \frac{dV}{dt} = \dot{V} + \omega \times V \quad (4)$$

Donde:

V=velocidad lineal del avión

ω = velocidad angular.

El vector aceleración en el CG del avión es:

$$\vec{a} = \left. \frac{d\vec{v}(t)}{dt} \right|_E = \left. \frac{d\vec{v}(t)}{dt} \right|_B + \omega \times \vec{v}(t) \quad (5)$$

Y expresando en término de los vectores unitarios

$$\vec{v} = U\hat{I} + V\hat{J} + W\hat{K} \quad (6)$$

Derivando tenemos.

$$\left. \frac{d\vec{v}(t)}{dt} \right|_B = \dot{U}\hat{I} + \dot{V}\hat{J} + \dot{W}\hat{K} \quad (7)$$

Donde (7) es la aceleración instantánea del aeroplano con respecto al sistema de ejes, si denotamos a las componentes del vector aceleración como a_x , a_y , a_z respecto del sistema de ejes, tenemos las siguientes ecuaciones algebraicas.

$$a_x = \dot{U} - RV + QW \quad (8)$$

$$a_y = \dot{V} - PW + RU \quad (9)$$

$$a_z = \dot{W} - QU + PV \quad (10)$$

Donde U,V y W representan la rapidez del avión en sus tres ejes, mientras que PQR corresponden a las velocidades angulares en torno a los mismos ejes.

Si consideramos que los vectores de las fuerzas actúan en el aeroplano y son expresadas en términos de las componentes de $\sum X, \sum Y, \sum Z$ a lo largo de los ejes del sistema de referencia tenemos que:

$$\sum X = m(\dot{U} - RV + QW) \quad (11)$$

$$\sum Y = m(\dot{V} - PW + RU) \quad (12)$$

$$\sum Z = m(\dot{W} - QU + PV) \quad (13)$$

Si realizamos el mismo análisis para $\sum H$:

$$\vec{H} = h_x \hat{i} + h_y \hat{j} + h_z \hat{k} \quad (14)$$

Descomponiendo el vector de momentos angulares respecto al sistema de ejes tenemos las siguientes ecuaciones:

$$h_x = I_{xx}P - I_{xy}Q - I_{xz}R \quad (15)$$

$$h_y = -I_{xy}P + I_{yy}Q - I_{yz}R \quad (16)$$

$$h_z = -I_{xz}P - I_{zy}Q + I_{zz}R \quad (17)$$

Donde $I_{xx} = \sum dm(y^2 + z^2)$ nos representa el momento de inercia del avión en torno al eje X, de igual forma para I_{yy} e I_{zz} mientras que I_{xy} es el producto de inercia del avión para los ejes X e Y.

De manera similar como se hizo con (4) obtenemos la ecuación para magnitudes angulares

$$\left. \frac{d\vec{H}}{dt} \right|_E = \left. \frac{d\vec{H}}{dt} \right|_B + \vec{\omega} \times \vec{H} \quad (18)$$

Donde el término $\left. \frac{d\vec{H}}{dt} \right|_E$ nos representa la tasa de cambio del momento angular respecto al sistema de ejes de referencia fijo en la tierra y el término $\left. \frac{d\vec{H}}{dt} \right|_B$ representa la derivada del momento angular referido al sistema de ejes.

Remplazando (15) (16) y (17) en (18) obtenemos un sistema de ecuaciones para expresar la suma de momentos de todas las fuerza sobre el avión, el cual podemos descomponer de la siguiente manera:

$$\sum X = \frac{dH_x}{dt} = I_{xx}\dot{P} + QR(I_z - I_y) - I_{xz}(\dot{R} + PQ) \quad (19)$$

$$\sum Y = \frac{dH_y}{dt} = I_{yy}\dot{Q} + PR(I_x - I_z) + I_{xz}(P^2 - Q^2) \quad (20)$$

$$\sum Z = \frac{dH_z}{dt} = I_{zz}\dot{R} + PQ(I_y - I_x) - I_{xz}(QR + \dot{P}) \quad (21)$$

Las ecuaciones (11) (12) y (13) así como (19) (20) y (21) nos describen el movimiento del avión en función de los momentos y fuerzas ejercidas en él. Lo que se obtiene es un sistema de seis ecuaciones diferenciales no lineales con 6 incógnitas.

Para comprender mejor este sistema de ecuaciones se realizan las siguientes acciones:

- Linealizar las ecuaciones, describiendo estas el movimiento que efectúa el avión cuando ha sido sometido a una pequeña perturbación a partir de una determinada posición de equilibrio o compensación.
- Separar los movimientos del avión, esto significa que los movimientos que ejecuta el avión en el plano de simetría no influyen en la creación de movimientos de guiñada o alabeo y viceversa, mas coloquialmente diríamos que el cambio en el ángulo de los alerones o del timón de la dirección no originan movimientos en el plano de simetría.

Por lo que si $V=P=R=0$ se simplifica el sistema de ecuaciones, no conteniendo ningún término referente al alabeo o guiñada y describiendo únicamente los movimientos en el plano de simetría que corresponden perturbaciones en este plano.

$$\sum F_x = m(\dot{U} + WQ) \quad (22)$$

$$\sum F_z = m(\dot{W} - UQ) \quad (23)$$

$$\sum F_y = \dot{Q}I_y \quad (24)$$

El sistema anterior suele denominarse ecuaciones del movimiento longitudinal.

VII Características de los aviones empleados: Sr. Falcón y Hobbystar 60

Como plataforma de pruebas se usó un aeromodelo de tipo entrenador, en la fase inicial del desarrollo se contó con un avión de este tipo, llamado Sr. Falcon (ver Fig.86) tenía poco espacio en cabina para albergar la electrónica, sin embargo el modelo es bastante estable y fácil de volar, este fue utilizado en las lecciones iniciales de vuelo.

Posteriormente se usó el aeromodelo Hobbistar 60MKIII, el cual es del tipo entrenador, este último contiene más espacio en cabina para albergar la electrónica desarrollada, ambos aviones contienen estabilizador horizontal por lo cual son bastante estables.

Las Características que nos proporciona cada fabricante son las siguientes:

Tabla 7. Características Sr. Falcon y Hobistar 60

Características		
	Sr. Falcon	Hobbistar 60 MKIII
Envergadura de alas	1752 [mm]	1805 [mm]
Área de Alas	52[dm ²]	57 [dm ²]
Peso	2.8 – 3.2 [kg]	3.2 - 3.6[kg]
Carga Alar	Dato No Proporcionado	56-63 g/dm ²
Longitud del Fuselaje	1346 [mm]	1400[mm]
Motor Requerido	2 tiempos 0.35 in ³ a 0.45 in ³	2 tiempos 0.60 in ³
Radio Control Requerido	R/C de 4 Canales con 4 Servos	R/C de 4 Canales con 4 Servos



Figura 81. Dimensiones de los Aviones Sr. Falcón y Hobbistar 60.

Debido a las características de estabilidad que contienen este tipo de aviones son la mejor opción para probar la electrónica asociada a la telemetría y control automático.

6.1 Aviónica.

La Aviónica se refiere a todos los elementos tecnológicos (eléctrico- electrónicos) aplicados a la aeronáutica.

Esta es la pieza clave de todas las aeronaves modernas. Las funciones más vitales de cualquier aeronave, civil o militar, están plenamente ligadas con dispositivos electrónicos. Al igual que los motores de un avión y su fuselaje, la aviónica es una de las partes más caras de la aeronave [18].

6.2 Instrumentación aeronáutica.

Un factor muy importante para que una aeronave sea segura es tener una buena coordinación entre la aeronave y el piloto, o la estación de control terrena, todo esto se lleva a cabo mediante la correcta aviónica, haciendo imprescindible lo siguiente: un sistema de comunicaciones, sistemas de navegación e instrumentos de vuelo así como un piloto automático.

La misión principal de los instrumentos de a bordo es proporcionarnos la información correcta y necesaria tal que podamos mantener al avión bajo control en cualquier condición de vuelo [23].

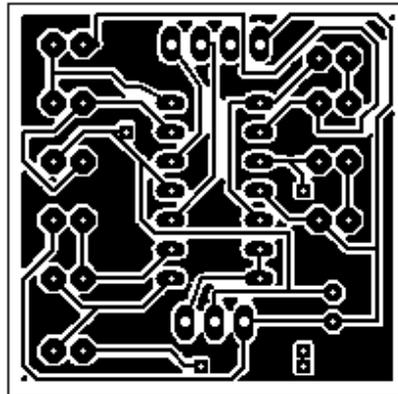
Con una correcta instrumentación se consigue controlar el vuelo y monitorear el estado de algunas variables del aeromodelo.

Anexos

Lista de Materiales

Lista de Partes Tubo de Pitot		
C1	0.1uf	Capacitor
MPX2010		Sensore
R1	10	Resistor
R2	10k	Resistor
R3	10	Resistor
R4	10k	Resistor
R5	1k	Resistor
R6	1k	Resistor
R7	50k	Resistor
R8	200k	Resistor
R9	50k	Resistor
R10	200k	Resistor
SV1		Resistor
iC1	OPAMP-TL084	OPAMP

Máscaras de Componentes



Ubicación de Componentes Vista Superior

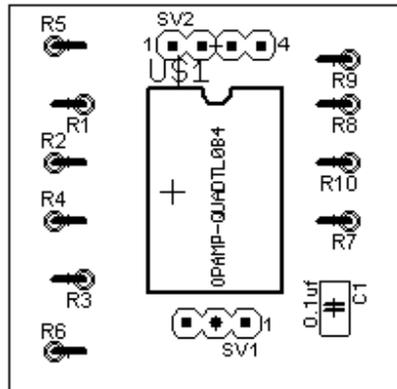
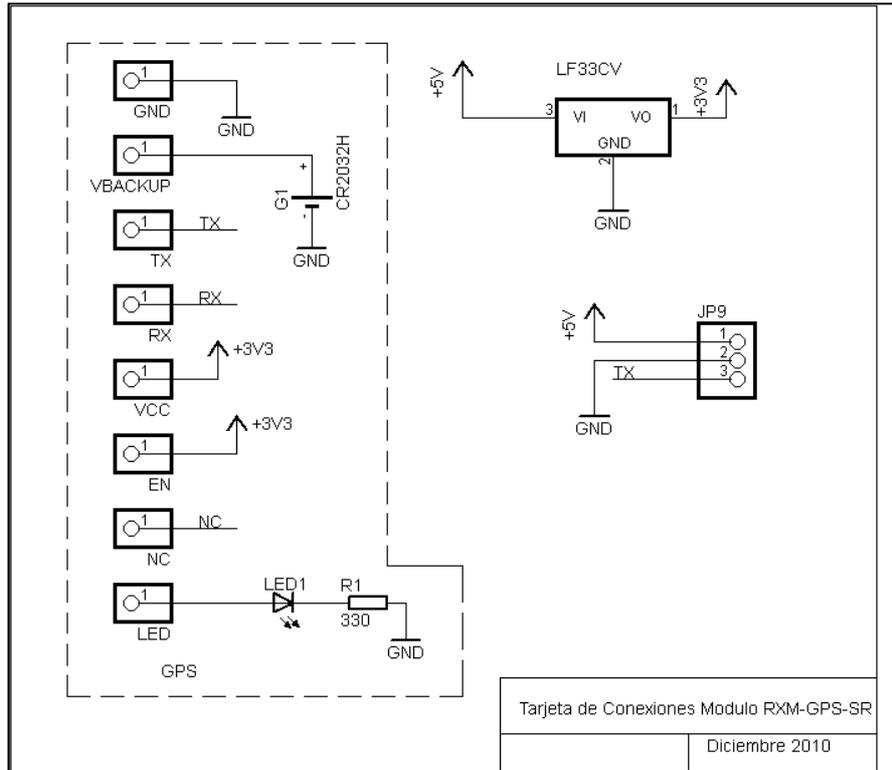


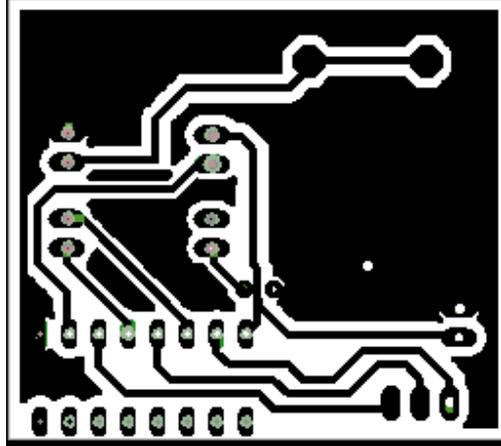
Diagrama esquemático tarjeta GPS.



Lista de Materiales

Lista de Partes				
Parte	valor	Dispositivo	Encapsulado	Libreria
EN		PINHD-1X1	1X01	pinhead
G1	CR2032H	CR2032H	CR2032H	battery
GND		PINHD-1X1	1X01	pinhead
JP9		PINHD-1X3	1X03	pinhead
LED		PINHD-1X1	1X01	pinhead
LED1		LED3MM	LED3MM	led
LF33CV		78L05Z	TO92	linear
NC		PINHD-1X1	1X01	pinhead
R1	330	R-EU_0204/5	0204/5	rc1
RX		PINHD-1X1	1X01	pinhead
TX		PINHD-1X1	1X01	pinhead
VBACKUP		PINHD-1X1	1X01	pinhead
VCC		PINHD-1X1	1X01	pinhead

Vista Superior Tarjeta GPS



Ubicación de Componentes

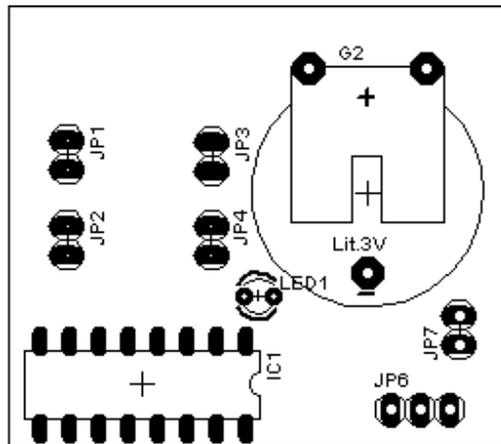
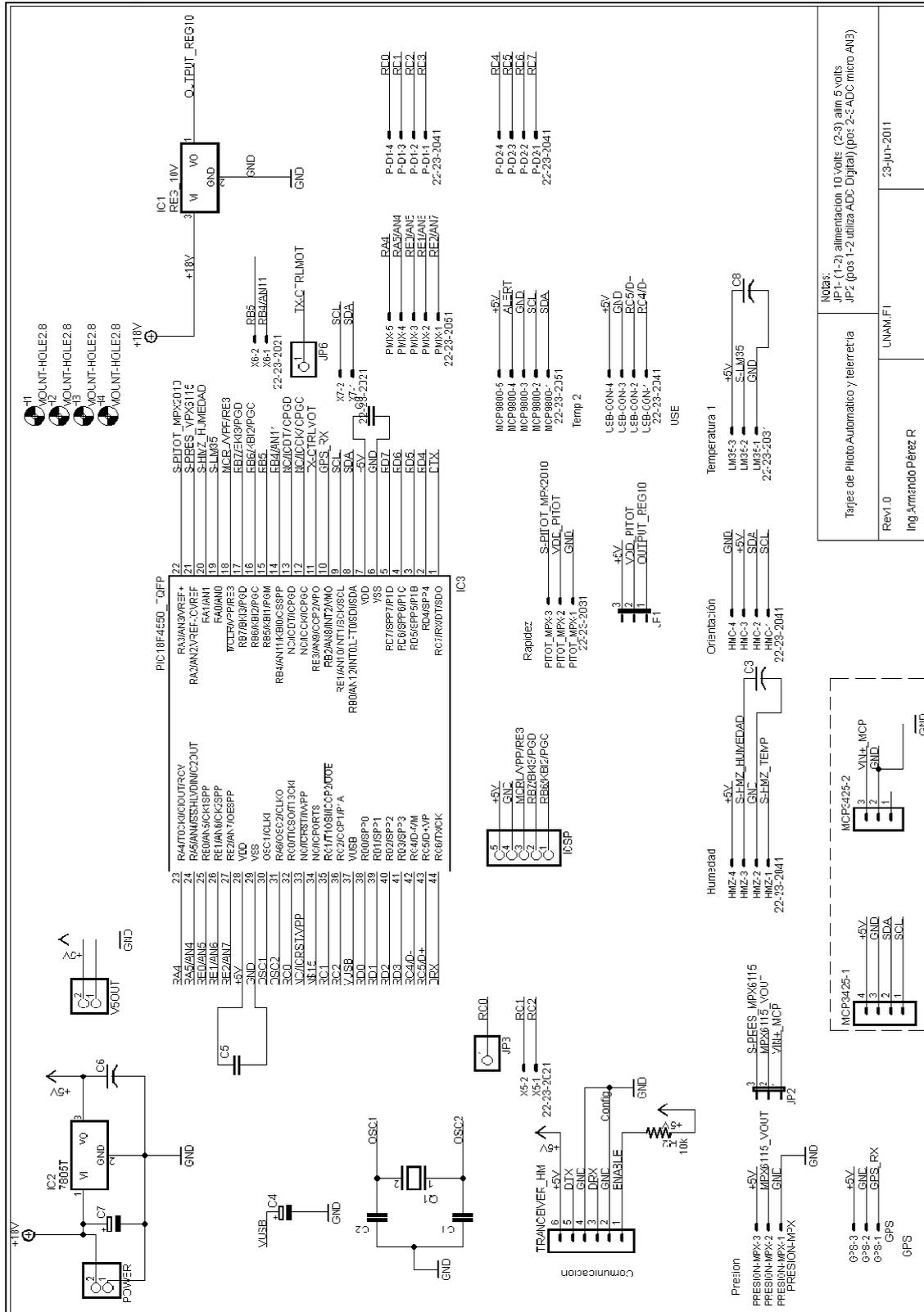


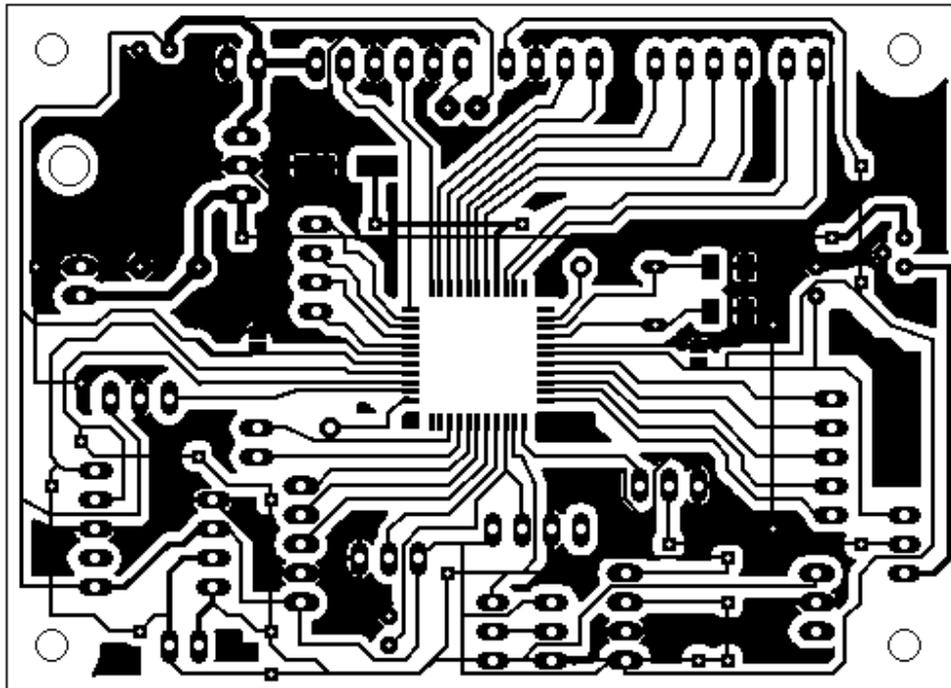
Diagrama esquemático tarjeta de Telemetría



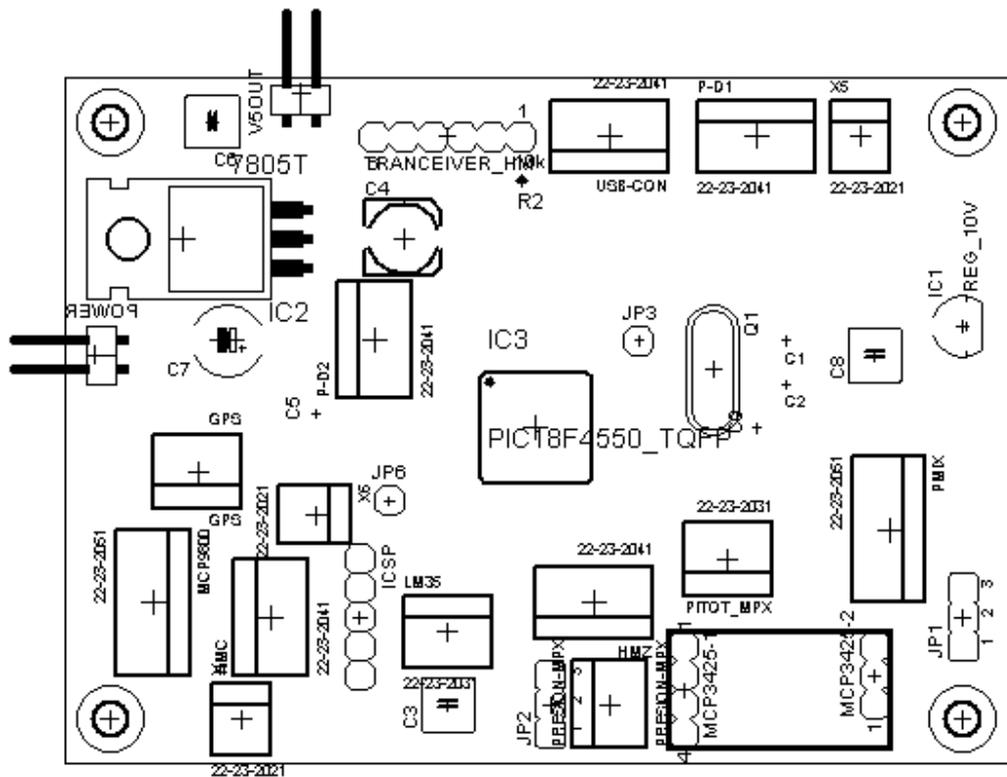
Lista de Materiales.

Parte	Valor	Dispositivo	Paquete	Libreria
C1		C-EUC1206K	C1206K	rc1
C2		C-EUC1206K	C1206K	rc1
C3		C-US025-050X050	C025-050X050	rc1
C4		CPOL-EU153CLV-0605	153CLV-0605	rc1
C5		C-EUC0504	C0504	rc1
C6		C-US025-050X050	C025-050X050	rc1
C7		CPOL-EUE5-6	E5-6	rc1
C8		C-US025-050X050	C025-050X050	rc1
C9		C-EUC0504	C0504	rc1
GPS	GPS	22-23-2031	22-23-2031	con-molex
HMC	22-23-2041	22-23-2041	22-23-2041	con-molex
HMZ	22-23-2041	22-23-2041	22-23-2041	con-molex
IC1	REG_10V	78L05Z	T092	linear
IC2	7805T	7805T	T0220H	linear
IC3	PIC18F4550_TQFP	PIC18F4550_TQFP	TQFP44	microchip
ICSP		PINHD-1X5	1X05	pinhead
JP1		JP2E	JP2	jumper
JP2		JP2E	JP2	jumper
JP3		PINHD-1X1	1X01	pinhead
JP6		PINHD-1X1	1X01	pinhead
LM35	22-23-2031	22-23-2031	22-23-2031	con-molex
MCP3425-1		MA04-1	MA04-1	con-1stb
MCP3425-2		MA03-1	MA03-1	con-1stb
MCP9800	22-23-2051	22-23-2051	22-23-2051	con-molex
P-D1	22-23-2041	22-23-2041	22-23-2041	con-molex
P-D2	22-23-2041	22-23-2041	22-23-2041	con-molex
PITOT_MPX	22-23-2031	22-23-2031	22-23-2031	con-molex
PMIX	22-23-2051	22-23-2051	22-23-2051	con-molex
POWER		PINHD-1X2/90	1X02/90	pinhead
PRESION-MPX	PRESION-MPX	22-23-2031	22-23-2031	con-molex
Q1		XTAL/S	QS	special
R2	10k	R-US_0204MIO	0204V	rlcmios
TRANCEIVER_HM		MA06-1	MA06-1	con-1stb
USB-CON	22-23-2041	22-23-2041	22-23-2041	con-molex
V5OUT		PINHD-1X2/90	1X02/90	pinhead
X5	22-23-2021	22-23-2021	22-23-2021	con-molex
X6	22-23-2021	22-23-2021	22-23-2021	con-molex
X7	22-23-2021	22-23-2021	22-23-2021	con-molex

Vista Superior tarjeta de Telemetría.



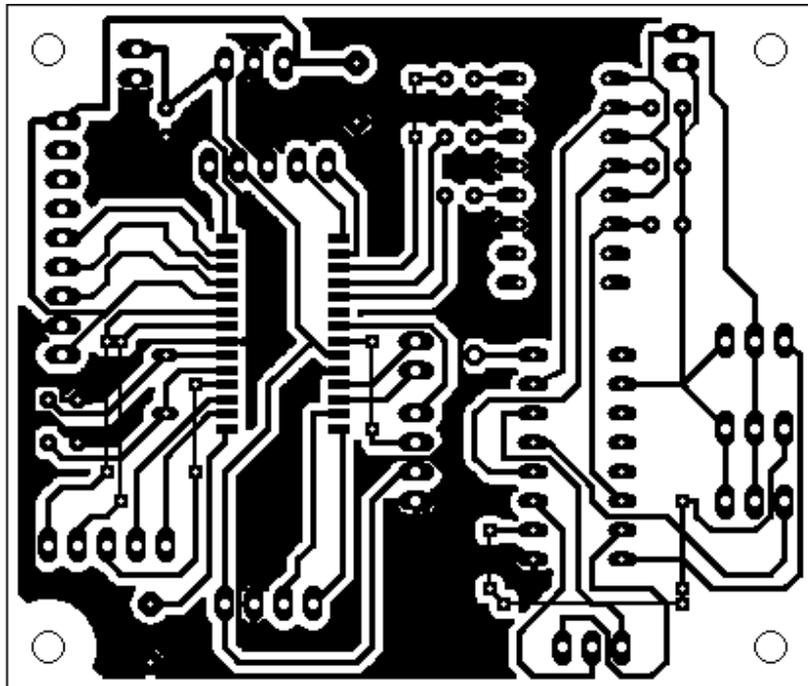
Ubicación de Componentes



Lista de Materiales.

Parte	Valor	Dispositivo	Paquete	Libreria
ACELEROMETRO_MMA7361		MA09-1	MA09-1	con-1stb
C1		C-EU025-025X050	C025-025X050	rc1
C2		C-EU025-025X050	C025-025X050	rc1
C3	.1uF	C-EU025-025X050	C025-025X050	rc1
C4	2.2uF	CPOL-EUE5-5	E5-5	rc1
C5	2.2uF	CPOL-EUE5-4	E5-4	rc1
HMC		PINHD-1X4	1X04	pinhead
IC1	74LS157N	74LS157N	DIL16	74xx-eu
IC2	LF33CV	7805TV	TO220V	linear
IC3	PIC18F2550_28w	PIC18F2550_28w	S028W	microchip
ISCP		PINHD-1X5	1X05	pinhead
JP3		PINHD-1X2	1X02	pinhead
JP4		PINHD-1X1	1X01	pinhead
JP6		PINHD-1X4	1X04	pinhead
JP7		PINHD-1X5	1X05	pinhead
OK5	PC847	PC847	DIL16	optocoupler
Q1		XTAL/S	Q5	special
R1	10k	R-EU_0204/2V	0204V	rc1
R2	10k	R-EU_0204/2V	0204V	rc1
R3	10k	R-EU_0204/2V	0204V	rc1
R5	330	R-EU_0204/2V	0204V	rc1
R6	330	R-EU_0204/2V	0204V	rc1
R7	330	R-EU_0204/2V	0204V	rc1
SV1		MA03-1	MA03-1	con-1stb
SV2		MA03-1	MA03-1	con-1stb
SV3		MA03-1	MA03-1	con-1stb
SV4		MA03-1	MA03-1	con-1stb
VBAT		PINHD-1X2	1X02	pinhead
VCTRL		PINHD-1X2	1X02	pinhead

Vista Superior Tarjeta de Piloto Automático.



Ubicación de Componentes Piloto Automático

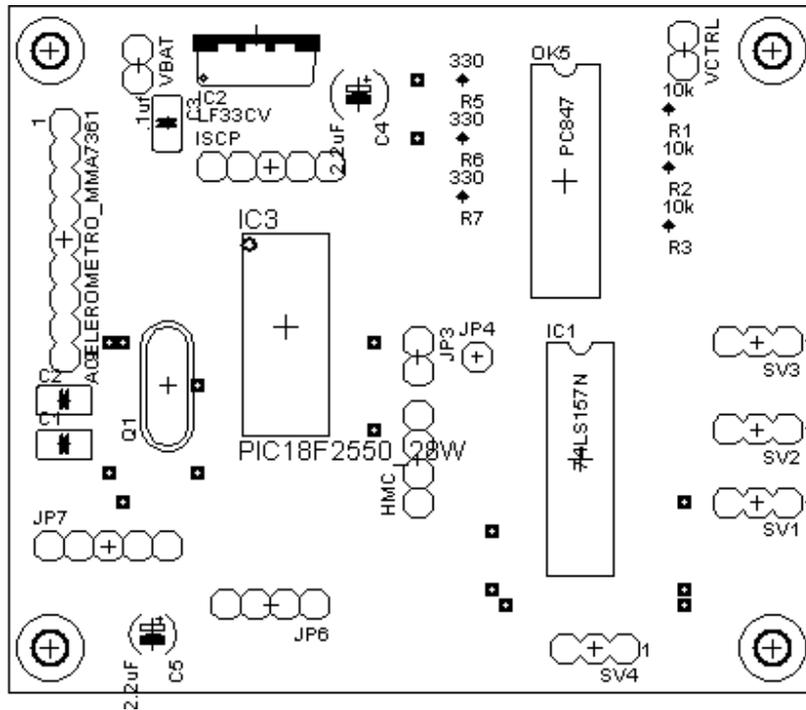
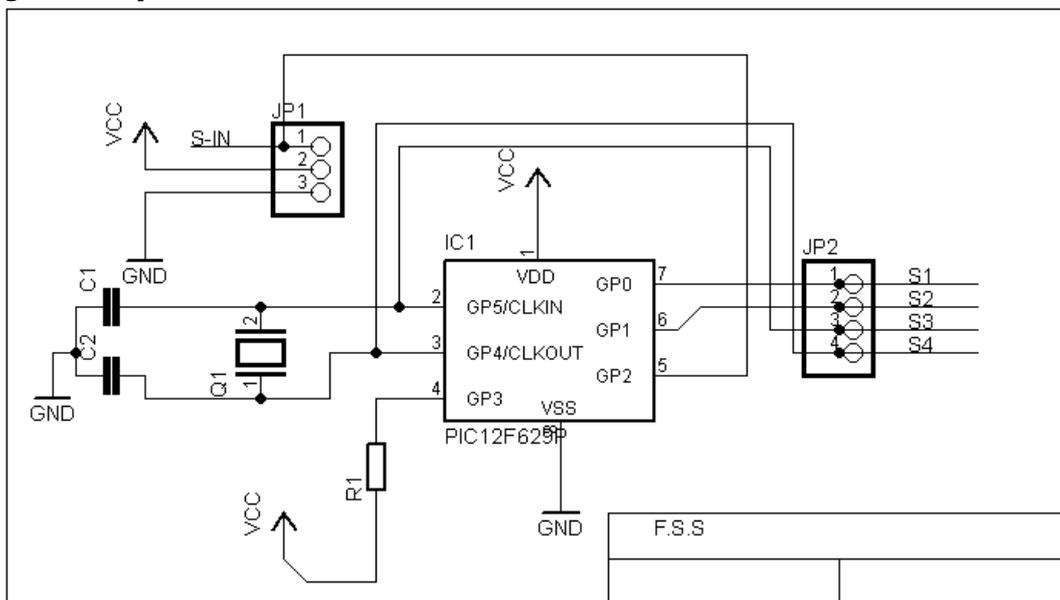


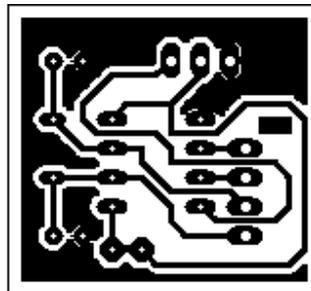
Diagrama Esquemático F.S.S



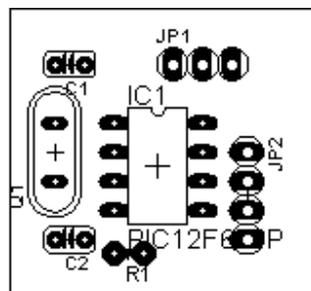
Lista de Materiales

Parte	valor	Dispositivo	Paquete	Libreria
C1		C-EU025-024X044	C025-024X044	rlcmios
C2		C-EU025-024X044	C025-024X044	rlcmios
IC1	PIC12F629P	PIC12F629P	DIL8	microchip
JP1		PINHD-1X3	1X03	pinhead
JP2		PINHD-1X4	1X04	pinhead
Q1		XTAL/S	QS	special
R1		R-EU_0204/2V	0204V	rlcmios

Vista Inferior F.S.S.



Ubicación de Componentes F.S.S.



Anexo II

Rutinas de Programación Tarjeta de Telemetría

```
////////////////////////////////////
////
//// Adquisición de datos y Posición Abordo del UAV
////
////////////////////////////////////
#include <18F4550.h>
#DEVICE ADC=10
#fuses
HSPLL,NOMCLR,NOWDT,NOPROTECT,NOLVP,NODEBUG,USBDIV,PLL5,CPUDIV1,VREGEN,NOI
CPRT
//#fuses NOMCLR
#use delay(clock=48000000)
#use rs232(baud=9600,/*ERRORS,*/
xmit=PIN_C6,rcv=PIN_C7,STREAM=COM_HM,bits=8) //Incluir el erro de
transmision en la tesis
#use rs232(baud=9600,/*ERRORS,*/ FORCE_SW
,xmit=PIN_B3,rcv=PIN_B2,STREAM=GPS,bits=8) //pagina 250 del
datashet
#use i2c(Master, sda=PIN_B0, scl=PIN_B1)
#priority RDA
#use fast_io(B)
#include <stdlib.h>
#include <string.h>
#include <input.c>
#include <variables_rda.h>
#include <variables_ad.h>
#include <variables_gps.h>
#define Device_SDA PIN_B0
#define Device_SLC PIN_B1
#define HMC6352_I2C_WRITE_ADDRESS 0x42
#define HMC6352_I2C_READ_ADDRESS 0x43
#define HMC6352_GET_DATA_COMMAND 0x41
static char command;
#define MCP9800_I2C_WRITE_ADRESSSS 0x90
#define MCP9800_I2C_READ_ADDRESS 0x91
char aux;
int8 byteMSB=0;
int8 byteLSB=0;
///Constantes Definiciones para el uso del buffer Usados en la interfaz
serial RS-232
#priority rda,ext
#int_rda
void serial_isr(void){
    if(kbhit(COM_HM)){
        command=getc(COM_HM);
        fprintf(COM_HM,"%c",command);
    }
} //Fin interrupcion rda

#INT_EXT
```

```

void ext_isr(void){
}
float heading_dgr;
unsigned long cont_hmc;
//subfunciones para el analisis del cheksum

int8 nmea_validateChecksum(char *strPtr);
unsigned int8 nmea_generateChecksum(char *strPtr);
char buftem[3];
i2c_start();
i2c_write(MCP9800_I2C_WRITE_ADRESSS);
i2c_write(0x00);
i2c_write(0x00);
i2c_read(MCP9800_I2C_READ_ADDRESS);
byteMSB=i2c_read();
byteLSB=i2c_read();
i2c_stop();
byteLSB=100*byteLSB/256;
fprintf(COM_HM,"byteMSB=%c byteLSB=%c\r\n ",byteMSB,byteLSB);
}

int16 HMC6352_read_heading(void)
{
int8 lsb;
int8 msb;
i2c_start();
i2c_write(HMC6352_I2C_WRITE_ADDRESS);//42hex
i2c_write(HMC6352_GET_DATA_COMMAND);//41hex letra A para recibir comando
cont_hmc=0;
for(cont_hmc=0;cont_hmc<6001;cont_hmc++) {delay_us(1);}
i2c_stop();
cont_hmc=0;
for(cont_hmc=0;cont_hmc<6001;cont_hmc++) {delay_us(1);}
i2c_start();
i2c_write(HMC6352_I2C_READ_ADDRESS);//43hex
msb = i2c_read();
lsb = i2c_read(0);
i2c_stop();
return((int16)lsb | ((int16)msb << 8));
}

void calibra_HMC(void ){
unsigned int8 i_hmc;
i2c_start();
i2c_write(HMC6352_I2C_WRITE_ADDRESS);
i2c_write(0x043);//43hex entramos en el modo de calibracion
cont_hmc=0;
for(cont_hmc=0;cont_hmc<11;cont_hmc++) {delay_us(1);}
printf("Rote durante 20 segundos dos veces \r\n>");
for(i_hmc=0;i_hmc<=40;i_hmc++)
{
printf("Tiempo Transcurido=%u\r\n>",i);
delay_ms(1000);
printf("\f");
}
i2c_write(HMC6352_I2C_WRITE_ADDRESS);
i2c_write(0x045);//sALIR DEL MODO DE CALIBRACION
delay_ms(15);//Tiempo requerido para completar la operacion
i2c_stop();
}

```

```

        printf("Calibracion completa\r\n",);
    }
    void leerCompas(){
        float heading;
            heading = HMC6352_read_heading();
            heading_dgr=heading/10;
    }
    void det_gps(void){
        fprintf(COM_HM, "\f"); //Limpiamos el buffer
    Aqu:
        if(!kbhit(GPS)){
            delay_ms(1);
            fprintf(COM_HM, "GPS_OFF");
            delay_ms(500);
            printf("\f");
            goto Aqu;
        }
        fprintf(COM_HM, "GPS_ON \r\n");
        delay_ms(1000);
        fprintf(COM_HM, "OK\r\n");
        fprintf(COM_HM, "Buscando...\r\n"); //Empieza Ciclo principal
    }
    void yinicbuff(void)
    {
        for(j=0;j<ylenbuff;j++) // Inicia a \0 cbuff ----
        { // Bucle que pone a 0
            ycbuff[j]=0; // caracteres en el
        }
        ybuff=0; // Inicializo el indice
    de siguiente caracter
    }
    void cadena_gps(void)
    {
        char c;
        ybuff=1;
        while(true) { // Si hay
            if(kbhit(GPS)){ // Si hay algo
                c=fgetc(GPS);
                switch(c){
                    case 0x0A:
                        yflagcommand=1;
                    break;
                    default:
                        ycbuff[ybuff++]=c;
                        //fin switch
                }
            }
            if(yflagcommand==1)
                break;
        }
    } //fin cadena gps
    void main_gps(void){
        yflagcommand=0;
        busca: if(kbhit(GPS)){ // Si hay algo pendiente .
            gpschar=fgetc(GPS); // lo descargo y ...
            if(gpschar=='$'){ //Comparamos con $ y
                ycbuff[0]=gpschar; //ycbuff[0]
            }
        }
    }

```

```

        cadena_gps(); // lo añ al buffer y
espera=0;
    } //else espera=1;
} //Fin kbhit gps
if(yflagcommand==1){ // Si Hubo datos
    if(ycbuffer[3]=='R'){
        irect=strlen(ycbuffer)-1; //Obtenemos la Longitud
for(i=0;i<irect;i++){ //Copiamos el Arreglo desde 0 hasta
    gpsbuffer[i]=ycbuffer[i]; //copiamos en rno
    }
    err_s=nmea_validateChecksum(gpsbuffer);
    fprintf(COM_HM,"*error=%u\r\n",err_s
if (err_s==0); {yflagcommand==0;}
for(i=0;i<irect;i++){
    fputc(ycbuffer[i],COM_HM
    }
    fprintf(COM_HM,"\r\n");
} //fin if caracter $
} // Fin If busqueda datos validos
if(yflagcommand==1){ //Borramos los ultimos datos
yinicbuffer(); //Limpiamos el Buffer
}
} //fin main gps
void analiza_com(void){
if(command=='I'){
    fprintf(GPS,"I\r\n");
    command=0;
}
if(command=='C'){
    fprintf(GPS,"C\r\n");
    command=0;
}
if(command=='X'){
    fprintf(GPS,"X\r\n");
    command=0;
}
}

void medidas(void);
void temperatura(void);
void horizonte(void);
void altimetro(void);
void humedad(void);
void pitot(void);
void print_sens(void);
void print_horizonte(void);

void main() {
    setup_adc(ADC_CLOCK_INTERNAL); //encendemos el ADC, Muestreo a 40
    setup_adc_ports(AN0_TO_AN6|VSS_VDD);
    enable_interrupts(int_rda);
    enable_interrupts(global);
    port_b_pullups(false);
    configura_9800();
while(true){ fprintf(COM_HM,"*error=");}
while(true){

```

```

        analiza_com();
        main_gps();
        cont_k++;
        if(cont_k==500)
            {cont_k=0;
             l++;

             if(l==8)
                 {   medidas();
                     l=0;
                     leerCompas();
                 }
            }
        } //Fin While
}/////////Fin Main
int8 nmea_validateChecksum(char *strPtr)
{
    int p;
    char c;
    unsigned int8 chksum;
    unsigned int8 nmeaChk;
    int8 flagValid;
    char hx[5] = "0x00";

flagValid = TRUE; //Empezamos con la bandera como verdadero, y la hacemos
falsa si las cosas no son correctas
if (strPtr[0] != '$' ) { flagValid = FALSE; }
    if (flagValid == TRUE)
        {
            c = strPtr[1]; //obtenemos el primer caracter
            chksum = c;
            p = 2;
while ( ( c != '*' ) && ( p < iresult-2 )          {
                c = strPtr[p]; // obtener el siguiente caracter
                if ( c != '*' ) { chksum = chksum ^ c; }
                p++;
            }
            hx[2] = strPtr[p];
            hx[3] = strPtr[p+1];
            hx[4] = 0x00;
            nmeaChk = atoi(hx);
            if ( chksum != nmeaChk ) { flagValid = FALSE; }
        }
    fprintf(COM_HM, "--chksum=%X--\r\n",chksum); /**<- solo
    return flagValid;
}
unsigned int8 nmea_generateChecksum(char *strPtr)
{
    int p;
    char c;
    unsigned int8 chksum;
    c = strPtr[0]; // get first chr  c = strPtr[0];
    chksum = c;
    p = 2;
while (c != 0x00) //Linea original while ( c != 0x00 )
    {
        c = strPtr[p]; // get next chr

```

```

        if (c != 0x00 ) { checksum = checksum ^ c; } //( c != 0x00 )
        p++;
    }
    return checksum;
}
void medidas(){
    setup_adc(ADC_CLOCK_INTERNAL); //encendemos el ADC, Muestreo a 40 MHz
    altímetro();
    temperatura();
    humedad();
    pitot();
    print_sens(); //Transmitimos los valores leídos
    setup_adc(ADC_OFF); //Apagamos el Convertidor A/D
}
void altímetro(){
    set_adc_channel(3);
    delay_us(10);
    cn2=read_adc();
}
void temperatura (void){
    set_adc_channel(0); //Activamos Ch4
    delay_us(10);
    cn0=read_adc(); //Guardamos la Lectura Sensor Temp
}
void humedad(){
    set_adc_channel(5);
    delay_us(10);
    cn1=read_adc();
}
void pitot(){
    set_adc_channel(6);
    delay_us(10);
    cn3=read_adc();
}
void print_sens(){
    fprintf(COM_HM, "%c0n%lu*c1n%lu*c2n%lu*c3n%lu*\r\n", cn0, cn1, cn2, cn3);
}
void borrarbuff()
{
    for(k=0;k<gpslen;k++)
    {
        // Bucle que pone a 0 todos los
        gpsbuff[k]=0; // caracteres en el buffer
    }
}
void copia_buffer(){
    for(j=0;j<ybuff;j++)
    {
        gpsbuff[j]=ybuff[j];
    }
}
}

```

Rutinas de Programación Tarjeta de Piloto Automático.

```
#include <18F2550.h>
#fusesHSPLL,NOWDT,NOPROTECT,NOLVP,NODEBUG,USBDIV,PLL5,CPUDIV1,NOVREGEN
#use delay(clock=48000000)
#use rs232(baud=9600, xmit=PIN_C6,rcv=PIN_C7,bits=8)
#use i2c(Master, sda=PIN_B0, scl=PIN_B1)
#include <math.h>
#use fast_io(B)
#priority timer1,rda
#define ELE_SERVO1 PIN_B5 //Elevador
#define AIL_SERVO2 PIN_B3 //Aileron/Alerones
#define RUD_SERVO3 PIN_B4 //Rudder/Timon
#define THR_SERVO4 PIN_B2 //Throttle/Acelerador
const int16 Ticks4Window = 65535;
const int16 Ticks4Minimum = 8403;
const int16 Ticks4Center = 18007;
const int16 Ticks4Maximum = 27611;
char cmd_pic;
static int16 Servo_PWM[3]={Ticks4Center,Ticks4Center,Ticks4Center};
static int8 Servo_Idx=0;
static int1 ELE_SERVO1_ON=1; //Con 20 Mhz 1tick =.2us
static int1 AIL_SERVO2_ON=1; //con 48 Mhz 1tick=.0833us
static int1 RUD_SERVO3_ON=1;
static int1 THR_SERVO4_ON=1;
static int1 flag_Phase;
static int16 Ticks4NextInterrupt=65535;
unsigned int16 reco;
int1 valid_command;
int8 i;
int8 j;
unsigned long k;
float lecX,lecY,lecZ;
float xs=0;
float ys=0;
float zs=0;
float x_off,y_off,z_off,z1_off;
float x_1,y_1,z_1;
const float voffset=1.2;
const float bicon=.00488758553274682307;
const float bicon33=.00322580645161290323;
const float sens=1.25; //Sensibilidad 1/800 mV
const float sens1=0.8; //Sensibilidad 800 mV
const float pil=3.141592654;
float Vx,Vy,Vz;
float Ax,Ay,Az,Angrx,Angry,Angrz;
float Angx,Angy,Angz;
int cont;
unsigned long flag_timer2=0;
unsigned long flag_1x;
unsigned long flag_2x;
unsigned long flag_3x;
float ADCX_OFF,ADCY_OFF,AZ_F;
```

```

int1 x_flag,y_flag;
const float xc1=-1.0;           //Constantes Usadas para  ajustar el nivel de
estabilidad del avion.
const float xc2=4.0;
const float xl=9.0;
const float x2=-3.0;
const float yc1=-2.0;
const float yc2=5.0;
const float yl=7.0;
const float y2=-7.0;
#int_rda
void serial_isr(void){
rcvchar=0x00;
if(kbhit()){
rcvchar=getc();           // lo descargo y ...
addcbuff(rcvchar);       // lo añ al buffer y ...
}}
#int_timer1
void timer1_isr(void){
if(flag_Phase==0){
if(Servo_Idx==0 && ELE_SERVO1_ON) output_high(ELE_SERVO1);
if(Servo_Idx==1 && AIL_SERVO2_ON) output_high(AIL_SERVO2);
if(Servo_Idx==2 && RUD_SERVO3_ON) output_high(RUD_SERVO3);
Ticks4NextInterrupt = 65535 - Servo_PWM[Servo_Idx];
set_timer1(Ticks4NextInterrupt);
}
if(flag_Phase==1){
if(Servo_Idx==0 && ELE_SERVO1_ON) output_low(ELE_SERVO1);
if(Servo_Idx==1 && AIL_SERVO2_ON) output_low(AIL_SERVO2);
if(Servo_Idx==2 && RUD_SERVO3_ON) output_low(RUD_SERVO3);
Ticks4NextInterrupt = 65535 - Ticks4Window + Servo_PWM[Servo_Idx];
set_timer1(Ticks4NextInterrupt);
if(++Servo_Idx>3) Servo_Idx=0;
}
++flag_Phase;
}           //Fin Interrupcion Timer1
void inicbuff(void){           // Inicia a  cbuff -----
-----
int k;
for(k=0;k<lenbuff;k++){           // Bucle que pone a 0 todos los
cbuff[ k ]=0x00;           // caracteres en el buffer
}
xbuff=0x00;           // Inicializo el indice de siguiente
}
int addcbuff(char c){           // Añ a cbuff -----
switch(c){
case 0x0D:           // Enter -> Habilita Flag para procesar
flagcommand=1;           // Comando en Main
break;
default:
cbuff[xbuff++]=c;           // Añ caracter recibido al Buffer
}
}
void delay_100us(void){
reco=0;
for(reco=0;reco<4801;reco++)
{ #asm

```

```

NOP
#endasm
}
}
void maximo(void){
for(i=0;i<4;i++)
Servo_PWM[0]=Ticks4Maximum;
}
void minimo(void){
for(i=0;i<4;i++)
Servo_PWM[0]=Ticks4Minimum;
}
void centro(void){
for(i=0;i<4;i++)
Servo_PWM[0]=Ticks4Center;
}
void cont_timer2(void){
}
void sensores(void){
j=0;
set_adc_channel(0);
for(j=0;j<11;j++) {delay_us(1);}
xs=read_adc();
j=0;
set_adc_channel(1);
for(j=0;j<11;j++) {delay_us(1);}
ys=read_adc();
}
void calibracion_zero(void){ //Poner el Acelerometro en posicion
horizontal.
cont=0;
set_adc_channel(0);
x_1=0;
for(cont=0;cont<50;cont++){
delay_us(11);
lecX=read_adc();
x_1+=lecX;
}
cont=0;
set_adc_channel(1);
y_1=0;
for(cont=0;cont<50;cont++){
delay_us(11);
lecY=read_adc();
y_1+=lecY;
}
cont=0;
z_1=0;
set_adc_channel(2); //par ala autocalibracion de Z se debe de
hacer de forma diferente.
for(cont=0;cont<50;cont++){
delay_us(11);
lecZ=read_adc();
z_1+=lecZ;
}
write_eeprom (0x01, x_1);
write_eeprom (0x02, y_1);

```

```

write_eeprom (0x03, z_1);
x_off = x_1/50;
y_off = y_1/50;
z1_off = z_1/50;
}
void calcula(void){
ADCX_OFF=xs-x_off;
ADCY_OFF=ys-y_off;
Z_F=zs-(z1_off-248);
}
Servo_PWM[2]=Ticks4Center;
if(heading_int>grados)
Servo_PWM[2]=Ticks4Center-7000;    if(Servo_PWM[2]<8700)
Servo_PWM[2]=8700;
if(heading_int<grados)
Servo_PWM[2]=Ticks4Center+7000;    if(Servo_PWM[2]>26000)
Servo_PWM[2]=26000;
}
void main(void) {
disable_interrupts(global);
setup_adc_ports(AN0_TO_AN2|VSS_VREF );           //referencia de 3.3 volts
setup_adc(ADC_CLOCK_INTERNAL);
setup_counters(RTCC_INTERNAL,RTCC_DIV_2);
setup_timer_0(RTCC_OFF);
setup_timer_1(T1_INTERNAL | T1_DIV_BY_1);
setup_timer_3(T3_DISABLED);
port_b_pullups(FALSE);
output_low(ELE_SERVO1);
output_low(AIL_SERVO2);
output_low(RUD_SERVO3);
output_low(THR_SERVO4);
enable_interrupts(int_rda);
set_timer1(Ticks4NextInterrupt);
set_timer2(65535);
enable_interrupts(int_timer1);
enable_interrupts(int_timer2);
enable_interrupts(global);
set_tris_a(0b00001111);
set_tris_b(0b00000001);
flag_accel = read_eeprom (0x00);
if(flag_accel==0x01){
delay_ms(1000);
calibracion_zero();           //Ajuste del offset del acelerometro
inicbuff();                 //Limpiamos el Buffer
delay_ms(1000);
}
centro();
x_off=read_eeprom (0x01);
y_off=read_eeprom (0x02);
while (true){
flag_timer2++;
if(flag_timer2>35535){
flag_timer2=0;
sensores();           //Obtenemos el Valor del Acelerometro Eje x,y,z
calcula();
if(flagrumbo==1){
sigue_rumbo();
}
}
}
}

```

```

}
}
if(ADCX_OFF<-2.0 ){
Servo_PWM[0]=Ticks4Center+7000;   if(Servo_PWM[0]>27600)
Servo_PWM[0]=27600;
}
if(ADCY_OFF<-2.0 ){
Servo_PWM[1]=Ticks4Center+6000;   if(Servo_PWM[1]>27600)
Servo_PWM[1]=27600;
}
flag_1x++;
if(flag_1x>32760){
flag_1x=0;
sensores();      //Obtenemos el Valor del Acelerometro Eje x,y,z
calcula();
}
if(ADCX_OFF>5.0 ){
Servo_PWM[0]=Ticks4Center-8000;   if(Servo_PWM[0]<8500)
Servo_PWM[0]=8500;
}
if(ADCY_OFF>5.0 ){
Servo_PWM[1]=Ticks4Center-6000;   if(Servo_PWM[1]<8500)
Servo_PWM[1]=8500;
}
flag_2x++;
if(flag_2x>500){
flag_2x=0;
sensores();      //Obtenemos el Valor del Acelerometro Eje x,y,z
calcula();
if(ADCX_OFF>-2.0 ||ADCX_OFF<5.0 ){
Servo_PWM[0]=Ticks4Center;
}
}
flag_3x++;
if(flag_3x>500){
flag_3x=0;
sensores();      //Obtenemos el Valor del Acelerometro Eje x,y,z
calcula();
if(ADCY_OFF>-2.0 ||ADCY_OFF<5.0 ){
Servo_PWM[1]=Ticks4Center;
}
}
flag_3x++;
if(flag_3x>500){
flag_3x=0;
sensores();      //Obtenemos el Valor del Acelerometro Eje x,y,z
calcula();
if(ADCX_OFF>-2.0 ||ADCX_OFF<5.0 & ADCY_OFF>-2.0 ||ADCY_OFF<5.0 ){
Servo_PWM[0]=Ticks4Center;
Servo_PWM[1]=Ticks4Center;
}
}
if(flagcommand==1)
{ procesa_comando();          // Si hay comando pendiente
printf("GradosPC=%lu  Cp_int=%lu
Cp_fl=%f\r\n",grados,heading_int,heading_dgr);
}
// de procesar ... lo procesa.

```

```
}  
}
```

Rutina de Programación Brújula Digital.

```
float    heading_dgr;    //variable para guardar el valor de orientación  
unsigned long cont_hmc;  
  
#define HMC6352_I2C_WRITE_ADDRESS 0x42    //Definición de Direcciones  
#define HMC6352_I2C_READ_ADDRESS  0x43    //De Lectura y escritura  
#define HMC6352_GET_DATA_COMMAND 0x41  
  
int16 HMC6352_read_heading(void)           //Función Para leer el Compass  
{  
    int8 lsb;    //Parte Baja  
    int8 msb;    //Parte Alta  
  
    i2c_start();    //Inicio de Comunicación I2C  
    i2c_write(HMC6352_I2C_WRITE_ADDRESS); //42hex  
    i2c_write(HMC6352_GET_DATA_COMMAND); //41hex letra A para recibir comando  
  
    cont_hmc=0;  
    for(cont_hmc=0;cont_hmc<6001;cont_hmc++)    {delay_us(1);}  
    i2c_stop();  
    cont_hmc=0;  
    for(cont_hmc=0;cont_hmc<6001;cont_hmc++)    {delay_us(1);}  
  
    i2c_start();  
    i2c_write(HMC6352_I2C_READ_ADDRESS); //43hex Para indicar lectura  
    msb = i2c_read();    //lectura de la parte alta de la lectura  
    lsb = i2c_read(0);    //lectura parte baja  
    i2c_stop();    //Detenemos al comunicación I2C  
  
    return((int16)lsb | ((int16)msb << 8)); //Regresamos parte alta y baja y  
    realizamos un corrimiento  
}  
  
void leerCompas(){  
    float heading;  
    heading = HMC6352_read_heading(); //Obtenemos el Valor de  
    Orientacion  
    heading_dgr=heading/10;    //10 para obtener grados  
    decimales  
    printf(" *hmc%3.1f*\r\n", heading_dgr); //Mostramos Valor de  
    Orientacion  
}
```