



**UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA**

**DISEÑO Y CONSTRUCCIÓN DE UN VEHÍCULO AÉREO
NO TRIPULADO**

**MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL
ELECTRICISTA**

RODRIGO ERNESTO ASENJO PREISLER

**PROFESOR GUÍA:
MARCOS ORCHARD CONCHA**

**MIEMBRO DE LA COMISIÓN:
HÉCTOR AGUSTO ALEGRÍA
SEBASTIÁN ISAO PARRA TSUNEKAWA**

**SANTIAGO DE CHILE
ENERO DE 2011**

RESUMEN DE LA MEMORIA
PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELECTRICISTA
POR: RODRIGO ASENJO PREISLER
FECHA: ENERO DE 2011
PROF. GUIA: Sr. MARCOS ORCHARD CONCHA

“DISEÑO Y CONSTRUCCIÓN DE UN VEHÍCULO AÉREO NO TRIPULADO”

El objetivo general del presente trabajo es el diseño y construcción de un vehículo aéreo no tripulado (UAV, por sus siglas en inglés) que sea capaz de mantener su horizontalidad en forma autónoma, es decir sin necesidad de la intervención de un piloto humano; utilizando para ello sistemas de control embebido que incorporen como referencia la presión atmosférica y el diferencial de temperatura entre la atmósfera y la superficie de la tierra. Para dichos efectos, se diseñará y construirá un aeroplano estable, fácil de volar y con capacidad de transportar la electrónica y sensores necesarios; y se implementará un sistema de sensores e instrumentación apropiados para el vehículo y sistema de control electrónico.

Actualmente, el costo de operación de un UAV comercial es muy elevado y requiere de personal altamente capacitado para una adecuada manipulación y mantenimiento. La compleja electrónica y programación limitan sus usos de manera importante, incluso para misiones sencillas. Este proyecto busca desarrollar una plataforma funcional, confiable, sencilla de operar y que permita la incorporación de nuevos sensores o dispositivos para realizar trabajos futuros. La naturaleza del proyecto despierta la curiosidad de los estudiantes y brinda la oportunidad de promover la carrera y motivar áreas no explotadas en la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile.

El proyecto se divide en tres grandes etapas: diseño y construcción del aeroplano, el diseño y construcción de la electrónica de automatización y finalmente, la integración de la electrónica y el aeroplano. Para ello se realizó un estudio de los sensores que actualmente son utilizados para alimentar al controlador con las distintas variables que permiten manipular al UAV. También se estudiaron las plataformas aéreas existentes y sus configuraciones aerodinámicas dependiendo del uso al que están destinadas. De esta forma fue posible identificar los sensores y la plataforma aérea necesaria para cumplir los objetivos planteados.

Los objetivos planteados para el prototipo del aeroplano derivaron en un “ala alta” de envergadura cercana a los 3[m] en configuración pusher. La planta propulsora elegida es un motor a explosión de 30[cc]. Los materiales utilizados en la construcción son básicamente fibra de vidrio, poliestireno expandido y madera de balsa.

Los sensores incorporados en la electrónica de automatización son: termocuplas infrarrojas para el control de la estabilización horizontal; GPS para la localización en el espacio; sensor de presión absoluta para determinar la altura de vuelo; y un sensor de presión diferencial para la velocidad de vuelo. El microcontrolador utilizado es un PIC24FJ128GA006 de gama media/alta.

Los resultados obtenidos demuestran que un estudiante puede diseñar y construir un UAV en la Facultad. El aeroplano mostró ser una plataforma estable, robusta y fácil de controlar. La electrónica probó que no es necesario contar con presupuesto ilimitado para obtener datos que permitan implementar un controlador para un UAV.

Se destaca que el trabajo de Memoria involucró todos los aspectos básicos de un proyecto de ingeniería, desde la conceptualización hasta la implementación del UAV. Dentro de la electrónica existen ciertos elementos que deben ser mejorados con el fin de obtener un vuelo 100% autónomo. Para trabajos futuros se deja entonces a disposición del estudiantado una plataforma de desarrollo de sistemas de control y posicionamiento.

Agradecimientos

Quiero comenzar dándole las gracias a mi familia por haberme apoyado durante todos estos años de difícil estudio. Aún cuando muchas veces no comprendían absolutamente nada de lo que estaba haciendo, seguían demostrando interés y preocupación. A mi hermana, por corregir mi ortografía y redacción.

A Marcos Orchard, mi profesor guía, que desde un comienzo se la jugó con todo para que mi proyecto personal de construir un UAV se pudiera convertir en mi trabajo de título. Me consiguió los fondos para financiar la construcción del prototipo y siempre me dio ánimo, aún en los momentos más difíciles.

A Don Francisco Brieva, Decano de la Facultad, que conociendo los riesgos implícitos en el diseño y construcción de un vehículo aéreo no tripulado, confió en nosotros, en que éramos capaces de sacar el proyecto adelante y así, demostrar el compromiso de la Facultad para con sus estudiantes.

A Isao “Zuneka” Parra por aceptar ser parte de mi comisión examinadora. Sin sus infinitos consejos y ayudas durante todo el proceso de diseño y pruebas de la electrónica, difícilmente habría logrado los resultados obtenidos. Gracias por acompañarme a los vuelos de prueba y darme ánimo en momentos en que estos no salieron bien.

A Paul Vallejos, que aunque sumido en su tesis de Doctorado, se hacía el tiempo y con su enfoque metódico, estudioso y paciente descubría las fallas en mis diseños electrónicos, programas y otros detalles que yo pasaba por alto o simplemente no comprendía y me aportaba valiosas ideas.

A mi amigo Gabriel “Gmoreno” Moreno por el infaltable apoyo en momentos complicados de la carrera. Por todos esos trabajos, tareas y presentaciones hechos con una maestría que nunca dejaba de asombrarnos. Por los almuerzos a las 12:30 hrs. y las correspondientes “órbitas” alrededor de la

Facultad. Porque logramos vencer todas las “conspiraciones” y terminar la carrera!

A mis amigos Ignacio “Herbert” Erbetta por aportar la seriedad necesaria en un grupo que a veces era pura “chacota” y Diego “Dmunoz” Muñoz por reírse siempre con nosotros.

A Felipe Tobar, Gonzalo Díaz y Aniella Descalzi. A todos los del semestre Beta, Beta’ y Beta”.

A Alejandro “Alexander Electricfield” Ehrenfeld, por haber confiado en mi para realizar diversos proyectos, en los que pude aprender muchísimo.

A los profesores “gurús” que tuve y que me enseñaron aspectos de la ingeniería más interesantes que calcular transformadas de Laplace e invertir matrices de 20x20 a mano.

A la Sra. Eliana “Sriliana” Monardes, que con su sonrisa y buen ánimo, incluso en los peores días, lograba transformar el ambiente del 3er piso del Departamento de Ingeniería Eléctrica, en el que constantemente se respiraban los humos que emanaban de miles de neuronas en estado de combustión.

A Vladimir “Vladimiri” Ovalle, coordinador del Taller Metal-Mecánico del Departamento, que me enseñó a trabajar con los tornos y fresadoras, y que con su experiencia y buena voluntad me ayudó a construir piezas que parecían imposibles.

Al profesor Javier Ruiz del Solar, Santiago “Chanti” Moreno y todos los funcionarios del Departamento, por haber hecho que mi paso por la carrera fuera mucho más ameno.

Finalmente, quiero agradecerle a “Sunny”, mi notebook, que ya en las últimas sigue dando lo mejor de sí!

Índice general

1. Introducción	1
1.1. Alcances del proyecto	1
1.2. Objetivos	2
1.2.1. Objetivos Generales	2
1.2.2. Objetivos Específicos	2
2. Contextualización	3
2.1. UAVs	3
2.1.1. Historia	3
2.1.2. Plataformas existentes	4
2.1.3. Elementos a considerar en el diseño de estrategias control de un UAV	5
2.2. Microcontroladores	8
2.2.1. Buses y Módulos	8
2.3. Sensores y sus aplicaciones	10
2.3.1. GPS	11
2.3.2. Giróscopos	12
2.3.3. Acelerómetros	12
2.3.4. Termocuplas infrarrojas	12
2.3.5. Sensor de presión absoluta	13
2.3.6. Sensor de presión diferencial	13
2.4. Integración Sensorial y Electrónica: construcción de un placa de circuitos integrados	13
2.5. Diseño de un aeroplano	16

2.5.1.	Las alas	16
2.5.2.	Los estabilizadores	17
3.	Diseño e Implementación del UAV	19
3.1.	Diseño del prototipo	19
3.1.1.	Electrónica	19
3.1.2.	Solución propuesta para la electrónica	20
3.2.	Diseño del aeroplano	38
3.2.1.	Solución propuesta para el diseño del aeroplano	39
3.2.2.	Estrategia de control	46
4.	Implementación y construcción del prototipo	49
4.1.	Electrónica	49
4.1.1.	Programas utilizados	50
4.1.2.	Esquemáticos	52
4.1.3.	Encapsulados de los componentes utilizados en el prototipo	60
4.1.4.	Ruteos	61
4.1.5.	Construcción	63
4.1.6.	Revisión de PCBs y soldado de los componentes	65
4.1.7.	Pruebas de funcionamiento de los módulos	66
4.2.	Construcción del aeroplano	71
4.2.1.	Programa utilizado	72
4.2.2.	Modelo computacional	72
4.2.3.	Materiales, técnicas y construcción	73
4.2.4.	Prueba del prototipo	86
4.3.	Integración de la electrónica en aeroplano	86
4.3.1.	Implementación del software	86
5.	Resultados y análisis	93
5.1.	Pruebas realizadas al aeroplano	93
5.2.	Sensores y electrónica	97
5.3.	Adquisición de datos durante vuelos de prueba en modo RC	98

5.4. Pruebas del sistema de control	100
6. Conclusiones y trabajo futuro	104
6.1. Conclusiones	104
6.2. Trabajo futuro	105

Capítulo 1

Introducción

Un vehículo aéreo no tripulado (UAV por sus siglas en inglés) se caracteriza por ser capaz de volar sin la intervención de un piloto humano, gracias a un sistema de pilotaje autónomo.

El diseño y construcción de un UAV comprende un gran desafío. Es necesario manejar conceptos relativos a la aerodinámica y tener experiencia en la construcción de aeronaves. El control necesita una variada gama de sensores mientras que la integración de dichos sensores requiere de amplios conocimientos de electrónica. Finalmente, el procesamiento de la información que se obtiene de los sensores para lograr manipular el UAV requiere del diseño e implementación de estrategias de control, cuya complejidad está en directa relación con el tipo de trayectorias a seguir durante la operación autónoma. Esta iniciativa califica ampliamente para ser considerada como un trabajo de Memoria para optar al título de Ingeniero Civil Electricista.

1.1. Alcances del proyecto

Actualmente, el costo de operación de un UAV comercial es muy elevado y requiere de personal altamente capacitado para una adecuada manipulación y mantenimiento. La compleja electrónica y programación limitan sus usos de manera importante, incluso para misiones sencillas. De ahí la importancia de este proyecto en el que se busca implementar una plataforma confiable, con capacidad de transportar una cámara y/o otros sensores a un costo muchísimo menor al ofrecido por las plataformas comerciales ya mencionadas.

Es importante precisar que dado que el despegue y aterrizaje son las etapas más riesgosas de un

vuelo, dichas etapas se ejecutarán en modo radio controlado (modo RC), es decir, manualmente. En el aire, el UAV será capaz de estabilizarse en el plano horizontal.

1.2. Objetivos

1.2.1. Objetivos Generales

1. Diseño y construcción de un vehículo aéreo no tripulado que sea capaz de mantener su horizontalidad en forma autónoma, es decir sin necesidad de la intervención de un piloto humano; utilizando para ello sistemas de control embebido que incorporen como referencia la presión atmosférica y el diferencial de temperatura entre la atmósfera y la tierra.
2. Dejar una plataforma funcional y sencilla de utilizar con una amplia documentación de respaldo. La capacidad de agregar dispositivos para realizar diferentes experimentos incentiva a futuros estudiantes a trabajar con esta plataforma.
3. Promoción de la carrera entre el estudiantado. La naturaleza del proyecto es atractiva para los estudiantes, tiene objetivos claros y despierta curiosidad.

1.2.2. Objetivos Específicos

1. Diseño y construcción de un aeroplano estable, fácil de volar y con capacidad de transportar la electrónica y sensores necesarios.
2. Lograr la integración de la electrónica en el aeroplano.
3. Implementación de un sistema de control electrónico y de sensores apropiados para el vehículo.

Capítulo 2

Contextualización

2.1. UAVs

2.1.1. Historia

UAV es el acrónimo de *Unmanned Aerial Vehicle*, que en español significa vehículo aéreo no tripulado. Estos vehículos, como su nombre lo dice, no son maniobrados por seres humanos sino que por computadoras. El no contar con un piloto en la aeronave, permite a los UAVs realizar tareas en las que se podría comprometer la integridad del piloto. En general, los UAVs deben cumplir misiones de reconocimiento y vigilancia, aunque ya hay disponibles versiones de combate (UCAV: *Unmanned Combat Aerial Vehicle*) que aún no se masifican. Algunos UAVs son operados remotamente y otros pueden realizar sus misiones de manera autónoma. Los primeros desarrollos datan de mediados de 1960 con el uso de “Drones” (o RPV: *remotely piloted vehicles*) para realizar misiones de reconocimiento, inteligencia y vigilancia.

En la Guerra del Golfo Pérsico (1990 - 1991) los drones jugaron un importante rol cuando EEUU tomó conciencia de las pérdidas humanas que tenían en misiones de reconocimiento. Con sólo 30 unidades y un total de 1560 horas de vuelo en 483 misiones, los Pioneer no pudieron dar la cobertura necesaria. Otros drones fueron utilizados en la operación Tormenta del Desierto, pero la falta de GPS y el corto alcance de la mayoría de las unidades hicieron presente la necesidad de contar con unidades capaces de realizar misiones más complejas y de mayor duración [1]. En 1998 el Global Hawk vuela por primera vez; con su envergadura alar de 40 metros, propulsado por una

turbina Rolls-Royce, capaz de realizar misiones de 40 horas de duración, con un techo de servicio de 20.000 metros y un costo de \$80 millones de dólares, representa al mayor exponente de los UAVs de reconocimiento. Cheng [2] presenta una tabla comparativa de los distintos UAVs con los que cuenta el Ejército Estadounidense.

2.1.2. Plataformas existentes

En la actualidad, es seguro indicar que cualquier vehículo aéreo se puede automatizar: helicópteros, aviones y hasta dirigibles. El tipo de plataforma que se va a utilizar depende únicamente de la finalidad que se persigue. Dependiendo del uso se pueden clasificar como:

1. De blanco - sirven para simular aviones o ataques enemigos en los sistemas de defensa de tierra o aire.
2. Reconocimiento - enviando información militar.
3. Combate - Combatiendo y llevando a cabo misiones que suelen ser muy peligrosas.
4. Logística - Diseñados para llevar carga.
5. Investigación y desarrollo - En ellos se prueban e investigan los sistemas en desarrollo.
6. UAVs comerciales y civiles - Son diseñados para propósitos civiles.

También es posible dividirlos por su techo de servicio y alcance de misión:

1. Handheld: 600 metros de altitud, 2[km] de alcance.
2. Close: 1500 metro de altitud, hasta 10[km] de alcance.
3. NATO (*North Atlantic Treaty Organization*): 3000 metros de altitud, hasta 50[km] de alcance.
4. Tactical: 5500 metros de altitud, hasta 160[km] de alcance.
5. MALE (*medium altitude, long endurance*): hasta 9000 metros de altitud y un alcance de unos 200[km].
6. HALE (*high altitude, long endurance*): sobre 9000 metros de techo y alcance indeterminado].
7. Hypersonic: alta velocidad, supersónico (Mach 1-5) o hipersónico (Mach 5+) unos 15000 metros de altitud o altitud suborbital, alcance de 200[km].

8. Orbital: en orbitas bajas terrestres (Mach 25+).

9. CIS-Lunar (región entre la atmósfera terrestre y la Luna): viaja entre la Luna y la Tierra.

Cada plataforma tiene ventajas y desventajas, como explica Ollero y Merino en [3]. La gran ventaja de los helicópteros y otros diseños de vehículos de despegue vertical es su maniobrabilidad, que es muy necesaria para aplicaciones robóticas. La posibilidad que brinda el mantener al vehículo en un vuelo estacionario es muy útil; sin embargo, el control de este tipo de vehículos es muy complejo, por lo que se requiere de experimentados pilotos para realizar pruebas. El vuelo totalmente autónomo requiere de complicadas estrategias de control no lineales.

Los dirigibles son habitualmente utilizados como plataformas para la toma de fotografías, dada su gran estabilidad. En caso de alguna falla de hardware, el dirigible caerá lentamente, a diferencia de otros vehículos como un helicóptero, el cual se precipita violentamente a tierra y las posibilidades de que el daño sea grave son muy altas. El manejo de un dirigible no requiere de un piloto muy capacitado, sin embargo, mantener el control sólo es posible si las condiciones meteorológicas son favorables: se requiere una baja velocidad del viento. Si bien es una plataforma interesante para algunas aplicaciones, las restricciones de movimientos y el difícil despliegue la hacen poco útil.

Los aviones también tienen restricciones de los movimientos que pueden realizar. La principal desventaja es la imposibilidad de realizar un vuelo estacionario, lo que reduce sus aplicaciones en el campo de la robótica. Sin embargo, el alcance y la duración de los vuelos pueden ser superiores a los vehículos de despegue vertical. Otra ventaja es el control: pilotear manualmente un aeroplano es relativamente sencillo, y por lo tanto, implementar un controlador que lo haga de manera automática no reviste tanta complejidad (en comparación con los vehículos de despegue vertical). Es por esto que los aeroplanos son los UAV's más comunes y pueden ser utilizados como plataformas de reconocimiento, mapeo de terreno, monitoreo y otros.

2.1.3. Elementos a considerar en el diseño de estrategias control de un UAV

La automatización de un vehículo aéreo tiene restricciones y desafíos adicionales a la automatización de un vehículo de tipo terrestre. Estas restricciones van de la mano con la capacidad máxima de carga para equipamiento extra y el máximo de consumo eléctrico permitido. Los

sistemas de control deben integrar la información proveniente de sensores y realizar el control de los servomotores, que son los actuadores que permiten controlar al vehículo.

Un aeroplano puede realizar giros en torno a sus ejes y también desplazarse, por lo tanto tiene 6 grados de libertad. Los movimientos en torno a sus ejes se clasifican como sigue:

- Cabeceo: ascenso o descenso.
- Alabeo: inclinación lateral.
- Guiñada: cambio en la dirección horizontal.

Para realizar estos movimientos se utilizan las distintas superficies de control que posee un aeroplano:

- Cabeceo: elevador.
- Alabeo: alerones.
- Guiñada: timón de cola.

De este modo las variables controladas serán: ángulo de cabeceo; ángulo de alabeo; ángulo de guiñada. Las variables manipuladas son los ángulos de inclinación de las superficies de control: elevador, alerones y timón de cola respectivamente.

En cuanto a la instrumentación necesaria en el UAV, es necesario mencionar que en la actualidad el campo de uso de los UAVs se ha expandido a ámbitos de aplicación civiles. Algunos ejemplos son el combate de incendios forestales de Ollero, Martínez de Dios y Merino en [4] y sistemas para la observación terrestre de Zhou y Zang en [5]. El principal enemigo de cualquier tecnología que migra del área militar al área civil son los costos. Este es el motivo por el cual gran cantidad de proyectos buscan abaratar los costos, integrando sensores de menor precisión con avanzado software implementado en plataformas más sencillas y de menor tamaño. En [6], Alison presenta una comparación de sensores.

La información a procesar proveniente de los sensores y el controlador para maniobrar el UAV dependen de la aplicación. La complejidad del controlador y la cantidad de información a procesar pueden resultar excesivas, lo que conlleva a disponer de un gran poder de procesamiento de datos. Este problema se puede abordar de dos maneras:

1. Enviar todos los datos a una estación en tierra que procese los datos y luego envíe las señales de control.
2. Realizar el procesamiento y control en el UAV.

Los retardos y la necesidad de contar con un *link* UAV-tierra perfecto convierten a la primera opción en la menos viable.

Los sensores para obtener los datos que permiten realizar el control que se utilizan en la actualidad son principalmente: GPSs; giróscopos; acelerómetros; compases digitales; magnetómetros; sensores de presión diferencial y sensores barométricos. Otro método utilizado para obtener datos de la actitud del aeroplano y así poder realizar un control adecuado, es la utilización de cámaras de video. Otro sistema que no se ve afectado por las aceleraciones a las que se encuentra sometido el aeroplano en vuelo es el uso de termocuplas infrarrojas que miden la temperatura del cuerpo al que estén apuntando.

Una vez que el microcontrolador obtiene los datos de los sensores, es el momento de procesarlos para realizar el control sobre el UAV. Las estrategias de control más utilizadas son controladores P (proporcional), PI (proporcional integral), PID (proporcional integral derivativo) y controladores difusos. Estos controladores son a menudo probados primero mediante simulaciones en programas como MATLAB® y finalmente son sintonizados en la vida real realizando métodos de prueba y error. Tal y como lo hacen Fou, Zhou y Xiong en [7], Scott, Nechyba, Ifju y Waszak en [9] y Abusleme en [13].

Otro componente infaltable en cualquier UAV es el eslabón (comunicación) RF a tierra que permite la comunicación bidireccional con el UAV. Este *link* permite conocer el estado del aeroplano (posición, velocidad, altura) y también brinda la posibilidad de modificar la misión una vez que ya se está ejecutando.

2.2. Microcontroladores

El explosivo desarrollo que han tenido los microcontroladores los han convertido en los candidatos perfectos para convertirse en el cerebro de los UAVs. Estos microcontroladores incorporan en un mismo encapsulado todas las unidades funcionales de un computador: CPU (unidad de procesamiento central), memoria y puertos de entrada/salida. Al tener todas las funcionalidades incorporadas necesitan muy pocos componentes externos para funcionar: un reloj y una fuente de energía. La elección de un microcontrolador depende de la aplicación: velocidad y capacidad de procesamiento, memoria, puertos de comunicaciones, conversores analógico/digital y temporizadores. Las casas proveedoras de microcontroladores más conocidas son Microchip™, Texas-Instruments™ y Atmel™ entre otras. El lenguaje de programación (*assembler*) es propio de cada proveedor, pero generalmente es una variación de C orientado a las distintas funcionalidades del microcontrolador. También es posible la programación en lenguajes de más bajo nivel. Si las funcionalidades y velocidades de procesamiento de un microcontrolador no son suficientes y no existen restricciones relacionadas con el tamaño y energía eléctrica disponible, siempre existe la posibilidad de utilizar plataformas más avanzadas como netbooks, notebooks o incluso computadores de altas prestaciones.

2.2.1. Buses y Módulos

Para la comunicación entre diversos dispositivos existen diferentes medios físicos y protocolos, como lo explica Parra en [32], estos están bien definidos para que exista compatibilidad entre los dispositivos.

2.2.1.1. SPI

Serial Peripheral Interface Bus o SPI es un protocolo de comunicación síncrona desarrollado por Motorola™. Está basado en *shift register* ya que dispone de una línea de entrada y una de salida y los datos van siendo desplazados mediante la señal de *clock* que es manejada por el *Master*. Está ideado para múltiples dispositivos que se seleccionan mediante una línea de *Chip Select*, que por cada dispositivo se requiere una línea de *Chip Select* diferente.

En total, requiere de cuatro líneas para su funcionamiento. Por este motivo también es conocido como *4-wires*.

2.2.1.2. I²C

Inter-Integrated Circuit o I²C es un protocolo de comunicación síncrona diseñado por Phillips™. Está basado un sistema de *pull-down* en que para transmitir se debe tirar la línea hacia abajo. Requiere solo de 2 líneas ya que una es el *clock* y la otra es la de *data*. La línea de *clock* solo es manejada por el *Master*. Se soporta varios *slave* donde cada uno debe tener una dirección diferente, estas direcciones son de 7 bits y son comunicadas mediante un paquete que define el protocolo. Cada vez que el *master* quiere comunicarse con un *slave* debe enviar la dirección, además si quiere leer o escribir en el dispositivo.

Este protocolo es muy versátil ya que permite agregar dispositivos sin tener que agregar líneas de control. También es conocido como *2-wires* ya que sólo posee dos líneas de datos.

2.2.1.3. UART

Universal Asynchronous Receiver-Transmitter o UART corresponde a una comunicación asíncrona que solo utiliza dos líneas, una para transmitir y otra para recibir. Es un medio de comunicación muy difundido por su sencilla forma de implementar. La velocidad de transmisión debe ser previamente definida por las dos partes.

En estas comunicaciones no existe jerarquía intrínseca para la comunicación, ya que por definición nadie controla la comunicación.

2.2.1.4. RS232

Define la interfase de comunicación para comunicaciones asíncronas, definiendo rango de voltajes y líneas. En su versión mas reducida, se implementa UART en este medio físico permitiendo comunicaciones a mayores distancia y robustez.

Es la capa física definida para los puertos seriales de los computadores, donde cada línea es transmitida con voltajes positivos y negativos.

2.2.1.5. USB

El *Universal Serial Bus* (USB) es un puerto que permite conectar diversos periféricos como teclados, mouses, cámaras fotográficas, scanners o impresoras a un computador sin necesidad de instalar programas especiales que manejen dichos periféricos.

2.2.1.6. Análogo

Algunos dispositivos tienen como salida un voltaje continuo dentro de un rango de operación. Para poder procesar la información de estos sensores se debe digitalizar esta señal. Algunos microcontroladores tienen módulos que realizan esta conversión. Estos módulos son llamados ADC (*Analog to Digital Converter*).

Existen diversos mecanismos para esta conversión, pero sus características fundamentales son: número de bits que corresponde a la resolución con la que pueden hacer la conversión y el tiempo de conversión que es cuanto tiempo requiere para dar el valor de la conversión.

2.2.1.7. PWM

La modulación por ancho de pulsos, PWM (*Pulse Width Modulation*), de una señal es una técnica en la que se modifica el ciclo de trabajo de una señal periódica, ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.

El ciclo de trabajo de una señal periódica es el ancho relativo de su parte positiva en relación con el período. Expresado matemáticamente:

$$D = \frac{\tau}{T} \tag{2.2.1}$$

Donde:

- D es el ciclo de trabajo
- τ es el tiempo en que la función es positiva (ancho de pulso)
- T es el período de la función

2.3. Sensores y sus aplicaciones

Los sensores de última generación utilizados en la mayoría de los proyectos de automatización de vehículos son los denominados MEMS (*Micro Electro Mechanical System*) que por el bajo costo y pequeño tamaño son la alternativa perfecta a los sensores de estándares militares. La desventaja que presentan estos sensores es el alto nivel de ruido que presentan sus lecturas. Un ejemplo de los

trabajos desarrollados con estos sensores y las distintas formas de mejorar las lecturas es presentado por Fu, Zhou, Xiong y Guo en [7] y por Jong-Hyuk, Wishart y Sukkarieh en [8].

En micro UAVs (MAVs) el espacio y capacidad de llevar carga limitan mucho los sensores que pueden transportar. Generalmente estos MAVs llevan consigo una cámara de video para realizar reconocimiento. Scott, Nechyba, Irfju y Waszak plantean la utilización de estas imágenes para realizar el control en [9]. La ventaja de la cámara por sobre otro tipo de sensores es la nula acción que tienen las aceleraciones sobre las imágenes. Las aceleraciones a las que se ve sometido un aeroplano en vuelo dificultan el uso de cualquier sensor MEMS de bajo costo. Además, los giróscopos, dada la naturaleza de su funcionamiento, tienden a acumular un error permanente que crece con el tiempo, lo que se traduce en la necesidad de realizar constantes recalibraciones, como lo muestra Moreno en [10].

Se describirán brevemente los sensores utilizados en proyectos similares.

2.3.1. GPS

GPS es la sigla de Sistema de Posicionamiento Global (*Global Positioning System*). Este sistema permite localizar el dispositivo GPS en cualquier punto del planeta indicando la latitud, longitud y altitud. El funcionamiento se basa en la triangulación de las señales provenientes de distintos satélites para así poder calcular la posición. El sistema GPS es administrado por el Departamento de Defensa de los Estados Unidos y su utilización no tiene costo alguno, sólo es necesario tener un dispositivo compatible con la red GPS.

En un UAV se utilizan principalmente para determinar la posición del vehículo en cualquier punto del globo terrestre. En una misión de reconocimiento, los puntos de interés se le indican al UAV utilizando coordenadas geográficas. A partir de estas coordenadas, posición actual y posición a la cuál se desea ir, el controlador puede definir una ruta que una los dos puntos y eventualmente realizar los cambios necesarios en la trayectoria para llegar al objetivo.

El dispositivo GPS se puede adquirir cómo una unidad con capacidad de cargar mapas y desplegarlos en una pantalla, o simplemente como un módulo que entrega mediante algún protocolo de comunicaciones la ubicación actual.

2.3.2. Giróscopos

El giróscopo es un sensor que permite conocer la velocidad angular con la que rota un cuerpo respecto a un eje. Utilizando giróscopos para medir las velocidades angulares en los distintos ejes del aeroplano y realizando una integración de estas mediciones es posible determinar cuanto ha rotado el cuerpo respecto a un eje y a una condición inicial.

Existen giróscopos netamente mecánicos; tienen un disco que gira a gran velocidad y que está alineado con el eje respecto al cual se quiere medir la rotación. También los hay del tipo MEMS. Los utilizados en UAVs son generalmente del tipo MEMS, ya que son de menor tamaño y mayor precisión. La interfaz con este tipo de sensores puede ser analógica (entregan un voltaje proporcional a la velocidad angular) o digital (SPI, UART o I²C).

2.3.3. Acelerómetros

El acelerómetro, como su nombre lo indica, es un sensor capaz de medir las aceleraciones a las que se encuentra sometido. En la tierra hace sus mediciones referenciadas a la aceleración de gravedad. En reposo un acelerómetro medirá la aceleración de gravedad en el sentido que corresponda y su función será la de un inclinómetro (medición de inclinación). Cuando el sensor se encuentra sometido a otras aceleraciones (un sistema no inercial) el resultado será más complejo de analizar. En este caso es usado para determinar la posición de un cuerpo, pues al conocerse su aceleración en todo momento, es posible calcular los desplazamientos que tuvo. Considerando que se conocen la posición y velocidad original del cuerpo bajo análisis, y sumando los desplazamientos medidos se puede determinar la posición. Las tecnologías actuales permiten incorporar en un mismo integrado acelerómetros que permitan medir aceleraciones en 3 ejes. En el integrado se incluye también la parte encargada de procesar las señales. El formato de salida puede ser analógico o digital.

2.3.4. Termocuplas infrarrojas

Las termocuplas infrarrojas son sensores que miden la temperatura del cuerpo al que estén apuntando. La señal que entregan puede ser analógica (un voltaje proporcional a la temperatura medida) o digital (utilizando algún protocolo de comunicación). Este sistema utiliza el diferencial de temperatura que existe entre el cielo y la tierra (la temperatura del cielo es menor a la de la tierra). Ubicando de manera inteligente un arreglo de estos sensores es posible estimar la actitud del UAV y

realizar la estabilización. Taylor, Bil y Watkins, en [11], sugieren que este método de estabilización es muy preciso y sencillo de implementar. Existen unidades comerciales para la estabilización de aeromodelos basados en estos sensores. A diferencia de los acelerómetros y giróscopos, este sistema no ve afectado su funcionamiento al estar sometido a aceleraciones. Sin embargo, el uso de estos sensores limita la utilización del UAV a condiciones meteorológicas favorables. Es importante destacar que en condiciones meteorológicas desfavorables, tampoco es posible utilizar métodos visuales para el control.

2.3.5. Sensor de presión absoluta

Este sensor se utiliza para determinar la altitud con respecto a una referencia. Esta referencia será generalmente el nivel del mar. Este sensor pertenece a la familia de los MEMS. La salida de estos sensores puede ser analógica o digital.

2.3.6. Sensor de presión diferencial

Utilizando dos sensores de presión unidos entre ellos, es posible determinar la presión diferencial entre dos puntos. En un aeroplano se utilizan para determinar la velocidad de vuelo, es decir, la relativa al aire. Los dos puntos que se utilizarán corresponden a una toma estática y a una dinámica. Es posible obtener estas presiones utilizando un tubo de Pitot (3.1.5).

2.4. Integración Sensorial y Electrónica: construcción de un placa de circuitos integrados

Existen varias formas de construir una placa de circuito integrados (PCB por sus siglas en inglés *Printed Circuit Board*), algunos son más sencillos que otros y por lo tanto más limitados en lo que respecta a la precisión y densidad de componentes. Un PCB en bruto (antes de soldar los componentes) se compone de un material dieléctrico recubierto por una lámina de cobre. Sobre esta lámina de cobre se realizan las pistas que unirán las señales de los diferentes componentes. La técnica más común para obtener las pistas consiste en retirar todo el cobre necesario para asilar las pistas entre ellas y dejan a la vista el material dieléctrico. Las placas pueden ser multicapa, esto es: tener más de una capa con cobre disponible para realizar las pistas. De esta forma, es posible soldar

componentes en ambos lados de una placa y uniendo pistas entre lado y lado (*layers*) utilizando vías. Una técnica muy utilizada para la implementación de las vías son remaches o cables que unen una señal de un layer con otro layer. En Chile es posible mandar a hacer placas con hasta 8 layers con la empresa Ciga [12], los layers que no se encuentran a la vista son solo para efecto de unión de señales y no para el soldado de componentes. El método más sencillo de fabricar un PCB consiste en dibujar las pistas sobre el cobre con un lápiz de tinta indeleble. El siguiente paso consiste en sumergir la placa en un líquido corrosivo (percloruro férrico) que corroe todo el cobre no dibujado. Luego, utilizando diluyente, se limpia el PCB con el fin de sacar la tinta y dejar las pistas de cobre al descubierto, se hacen los agujeros que sean necesarios y finalmente se soldan los componentes. Este método es el más sencillo y útil para placas de muy baja complejidad y reducido número de componentes. Con el lápiz y dibujando a mano alzada es difícil conseguir alta precisión. El segundo método consiste en diseñar, utilizando diversos programas computacionales, un esquemático del circuito a realizar y luego a partir de ese esquemático, posicionar los componentes y convertir los nodos en pistas. Este último proceso, convertir los nodos en pistas, es llamado “ruteo”. Al finalizar la etapa de ruteo se obtiene un modelo en el que se encuentran representados tanto los componentes como las pistas. Dependiendo de la complejidad del ruteo obtenido, del tamaño de las pistas y del tamaño de los componentes, es posible hacer el PCB utilizando distintos métodos:

- Impresión del ruteo y transferencia de éste por medio de calor a la placa de cobre. Se imprime el ruteo en una transparencia utilizando una impresora láser. Se coloca la impresión sobre la placa de cobre y utilizando una plancha caliente se transfiere el toner de la transparencia a la placa de cobre. Luego se sumerge la placa en percloruro férrico y se obtiene la PCB. Con este método y mucha práctica es posible obtener muy buenos resultados tanto para componentes *through hole* como SMDs (*Surface Mount Devices*) con una separación entre pads de mínimo 35[mil].
- Impresión del ruteo en una transparencia y transferencia utilizando una película fotosensible. Al igual que el método anterior, es necesario imprimir en la mejor calidad posible el ruteo en una transparencia. La placa de cobre es rociada con una sustancia fotosensible que reacciona a la radiación ultravioleta. La transparencia se ubica sobre la placa de cobre con la película de sustancia fotosensible y a continuación se revela de la misma forma que se revela una fotografía utilizando luz ultravioleta. De la película fotosensible sólo quedará en el cobre lo

correspondiente a las pistas. Luego se sumerge la placa en percloruro férrico y se obtiene la PCB. Con este método es posible obtener mejores resultados que utilizando la transferencia por calor, pero requiere de más práctica y los insumos son más costosos.

- Mecanizado utilizando una fresadora Control Numérico por Computador (CNC). Este proceso se denomina “prototipado rápido”. El ruteo se exporta a un programa que transforma los archivos Gerber generados por el ruteo a coordenadas y recorridos sobre la placa de cobre del cabezal de la fresadora. Este archivo es entonces exportado al programa controlador de la CNC. El resultado es una PCB de gran definición y de hasta 2 layers. Es llamado prototipado rápido por que el tiempo necesario para hacer una placa es muy poco pero el costo de producir una placa lo convierte en un método prohibitivo para la producción en masa.
- Mandar a hacer las PCB a una empresa especializada. Este último método es el más seguro y con el que se obtienen los mejores resultados. La PCB se hace a partir de la deposición de cobre sobre la placa de dieléctrico. El resultado son placas de hasta 8 layers (en Chile) de excelente precisión y acabado profesional. El inconveniente es el costo: el costo inicial de producir la primera placa es muy elevado y sólo se justifica si la placa es muy compleja, tiene más de 2 layers, se tiene un elevado presupuesto o se quiere realizar una producción en masa.

2.5. Diseño de un aeroplano

Dependiendo de las características de vuelo que se quiere conseguir existen una serie de factores a considerar en el diseño de un aeroplano: posición, ángulo de diedro, forma y perfil del ala; posición y configuración de los estabilizadores; y finalmente la configuración del fuselaje con la ubicación de la planta propulsora.

2.5.1. Las alas

Existen muchas clasificaciones en torno al ala de un avión. En la fase de diseño una de las primeras características que se debe definir es el perfil alar. Del perfil dependerán las cualidades de vuelo del aeroplano y éste se elige tomando en cuenta la estabilidad, maniobrabilidad y velocidad requeridos. Si bien existen miles de perfiles, es posible realizar una sencilla clasificación observando a grandes rasgos la forma:

- Fondo plano: es utilizado en aviones entrenadores dada su gran estabilidad y sustentación.
- Fondo cóncavo: muy sustentable, pero no adecuado para vuelos de velocidad.
- Simétrico: es el perfil que utilizan los aviones acrobáticos, las prestaciones en vuelo invertido y normal son muy similares.
- De velocidad: utilizado en aviones que requieren volar a grandes velocidades, es un perfil relativamente plano y que genera poca sustentación.

La ubicación de las alas en el fuselaje también definen la actitud del avión en lo que a su maniobrabilidad respecta:

- Ala alta: el ala va montada en la parte superior del fuselaje, se utiliza generalmente en aviones donde se requiere estabilidad y baja maniobrabilidad.
- Ala media: ala ubicada en el medio del fuselaje, brinda estabilidad y maniobrabilidad. Esta configuración es generalmente utilizada en aviones acrobáticos.
- Ala baja: ala montada en la parte inferior del fuselaje, muy poca estabilidad pero gran maniobrabilidad.

El ángulo que forman las alas con el plano se llama diedro. El diedro está estrechamente ligado con la estabilidad del aeroplano. Las alas pueden tener diedro positivo, negativo o simplemente ser planas. El ángulo puede presentarse: en el centro del ala, o sea el lugar donde se unen las dos semi-alas, que también equivale al centro del fuselaje; diedro en los extremos del ala; varios ángulos a lo largo del ala. Este ángulo tiende a equilibrar naturalmente el balanceo del avión. Esto se debe al fenómeno aerodinámico que se produce cuando una semi-ala esta horizontal y la otra inclinada: el ala horizontal producirá mayor sustentación que el ala inclinada, por lo que tenderá a subir hasta que la sustentación de ambas alas sea igual. De aquí se deduce que a mayor ángulo de diedro más estable es el aeroplano. Los aviones acrobáticos no tienen diedro, a diferencia de los entrenadores que tienen un diedro muy pronunciado.

2.5.2. Los estabilizadores

Según Abusleme en [13], los estabilizadores se pueden clasificar según su posición y su disposición con respecto al estabilizador vertical.

2.5.2.1. Clasificación de los estabilizadores horizontales según su posición

El estabilizador horizontal del avión puede ir delante o detrás de las alas. Ambas configuraciones presentan ventajas y desventajas, y se describen a continuación:

- Estabilizador en cola: Es la configuración típica utilizada en la mayoría de los aviones. Presenta la gran ventaja de darle estabilidad al avión. Sin embargo, tiene la desventaja de disminuir la sustentación total para elevar la nariz del avión, por lo que requiere mayor rapidez para despegar.
- Estabilizador Canard (estabilizadores en el morro del avión): Esta disposición permite elevar la nariz del avión mediante un aumento en la sustentación, lo que hace más fácil el despegue. También presenta grandes ventajas de maniobrabilidad. Sin embargo, tiende a ser mucho más inestable que la configuración convencional, por lo que requiere sistemas de control especiales.

2.5.2.2. Clasificación de los estabilizadores horizontales según su disposición con respecto a los verticales

Cuando el avión tiene una cola con estabilizadores horizontales y verticales en ella, éstos se pueden disponer de diferentes formas:

- Cola convencional: Es la más usada en aviones con cola y presenta a los estabilizadores horizontales como la parte final del fuselaje, y el estabilizador vertical sobre éste. Tiene la forma de una T invertida.
- Cola en V: Es un tipo de cola que tiene la forma de una V, con dos superficies de control en lugar de tres. Éstas combinan sus deflexiones para dar lugar al equivalente a las deflexiones de una cola convencional.
- Cola en T: Esta cola tiene forma de T y presenta la ventaja de estar alejada de las corrientes de aire producidas por el motor y las alas.

Capítulo 3

Diseño e Implementación del UAV

3.1. Diseño del prototipo

Requerimientos

Tomando como base los objetivos generales y específicos planteados anteriormente, es posible definir las características que darán forma al UAV. Dichas características definirán, en suma, los aspectos más significativos de la electrónica y el tipo de aeroplano a construir. Los requerimientos y diseño, tanto de la electrónica como del aeroplano, se estudian a continuación.

3.1.1. Electrónica

La electrónica es el cerebro del UAV, ésta es la encargada de procesar toda la información relativa al vehículo y ejecutar las correspondientes acciones de control y toma de decisiones. La información proviene tanto de sensores montados en el aeroplano como también de señales enviadas desde tierra mediante un *link* de radio frecuencia (RF). Como el control entre el modo autónomo y modo RC se maneja desde el radio control, la electrónica debe escuchar también las señales de salida del receptor del sistema RC. El sistema es de característica MIMO (*Multiple-input Multiple-output*), por lo que es necesario monitorear constantemente todas las variables que sean significativas desde el punto de vista de la estrategia de control. La selección de dichas variables determina, a su vez, el tipo de sensores a utilizar. Adicionalmente, es necesario mencionar que sin perjuicio de lo anterior se contempla también el monitoreo de otras señales con el fin de permitir la detección de anomalías

y estimar autonomía de vuelo. Las variables se detallan a continuación:

- Ángulo de alabeo.
- Ángulo de cabeceo.
- Posición geográfica (latitud y longitud).
- Rumbo.
- Altitud.
- Velocidad respecto al aire.
- Voltaje de baterías de alimentación de la electrónica, servos y receptor.

Dado que la electrónica debe ir montada en el UAV, el tamaño y la masa de ésta son críticas, debiendo ser lo más pequeña y liviana posible, lo que en definitiva fija parámetros importantes en el diseño del aeroplano. La electrónica también debe contemplar la posibilidad de expansión, esto es: la capacidad de integrar y controlar nuevos sensores y dispositivos, como por ejemplo una cámara fotográfica. Debe existir un puesto de control en tierra, en donde un operario pueda recibir y eventualmente enviar información de distintos parámetros relativos al vuelo del UAV y misión. El *link* inalámbrico debe estar dimensionado para lograr realizar misiones que le permitan al UAV alejarse varios kilómetros sin perder comunicación con el puesto de control en tierra.

3.1.2. Solución propuesta para la electrónica

Con los requerimientos claros se procede al diseño del prototipo en las etapas planteadas a continuación. En primer lugar, se selecciona el tipo de sensores a utilizar y a partir de las características de éstos se define el microcontrolador a utilizar.

3.1.2.1. Elección de los sensores

En el Capítulo 2 se exponen las distintas combinaciones de sensores utilizados en variados UAV's, así como también la función de cada uno. El número de sensores a utilizar y su precisión depende de las variables que se quieren medir y del presupuesto con el que se cuenta para la realización del proyecto. La incorporación de un giróscopo de 3 ejes y un acelerómetro de 3 ejes

conforman una IMU. El costo de una IMU de bajas prestaciones se escapa del presupuesto con el que se cuenta en este proyecto. La utilización de giróscopos y acelerómetros por separado implica lidiar con problemas relativos a las constantes descalibraciones de ambos sensores, y dada la naturaleza de esta memoria se optó por no incorporar estos sensores en el primer prototipo. Sin embargo, la incorporación de nuevos sensores es tomada en cuenta al momento de elegir el microcontrolador y por lo tanto, existe la posibilidad de ampliar la plataforma de sensores en trabajos futuros.

3.1.2.2. Medición de ángulos de alabeo y cabeceo

La estimación de estos ángulos se utilizan en la implementación del lazo principal de control de vuelo del UAV. Para la medición de los ángulos de cabeceo y alabeo se utiliza un arreglo de sensores infrarrojos de temperatura (termocuplas). Este sistema fue desarrollado por la NASA [14] para estabilizar satélites en órbita. La sencillez del funcionamiento y el hecho de que las mediciones no son afectadas por las fuerzas de aceleración que experimenta el aeroplano lo convierten en un sistema muy utilizado en UAVs de bajo costo e incluso como sistema de estabilización en aeromodelos de entrenamiento. Como es un sistema ampliamente utilizado es fácil encontrar buena documentación y esquemáticos ya probados. Si bien existe más de un fabricante de este tipo de sensores, los que mayor demanda tienen son los de la marca Melexis™.

Dentro de los modelos disponibles es posible encontrar: interfaces digitales o análogas, diferentes encapsulados y distintos voltajes de alimentación. Las interfaces digitales requieren que el procesador principal realice constantes consultas (I²C) o tomas de tiempo (PWM) para 4 ó 6 sensores distintos, lo que implica mucho tiempo de procesamiento. Además, dada la naturaleza de la aplicación, los sensores no estarán montados en la misma placa que el controlador -de hecho, estarán separados por una distancia que puede fácilmente llegar a los 20[cm]- lo que para el bus I²C está fuera de norma y no se garantiza su adecuado funcionamiento. Para superar el problema de las distancias es posible implementar una placa con un microcontrolador que realice las consultas de temperatura a los diferentes sensores y se comunique con el procesador principal utilizando el protocolo serial, pero esto implica utilizar un puerto serial extra en el procesador. Es por estos motivos que el tipo de interfaz elegido es el analógico. Existen procesadores que permiten realizar las conversiones de voltaje (análogo a digital) en sus pines de manera automática y de esta forma ahorrar uso del procesador. El voltaje de operación es el típico de un microcontrolador de gama media/alta: 3 - 3.3[V]. El encapsulado sólo difiere uno del otro en el tamaño (orden de milímetros),

por lo que se eligió el más económico. Considerando toda la información anteriormente presentada, se elige finalmente el Melexis™ MLX90247 mostrado en la Figura 3.1.1.

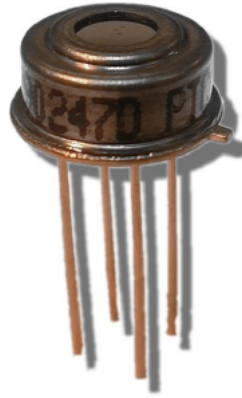


Figura 3.1.1: Ilustración de una termopila

Para medir un ángulo, por ejemplo el de alabeo, se debe utilizar un mínimo de dos sensores apuntando en dirección perpendicular al eje en torno al cual el aeroplano alabea y en sentidos opuestos. De esta manera, se mide la temperatura en ambos sentidos a los que apuntan las alas. El voltaje de salida del sensor está directamente relacionado con la temperatura medida, comparando ambas lecturas es posible determinar el ángulo de alabeo del aeroplano; si ambos voltajes son iguales se puede decir que el ángulo de alabeo es cero. La estrategia utilizada normalmente, consiste en utilizar la configuración de sensores de la Figura 3.1.2.

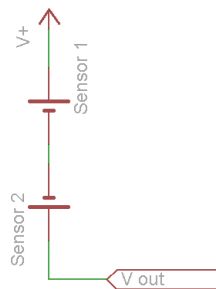


Figura 3.1.2: Configuración termopilas

Ambos sensores se conectan en cascada y con polaridades distintas. El voltaje de salida de ambos sensores estará encontrado dada la polaridad inversa en la que se encuentran conectados,

esto permite que con sólo la lectura de un voltaje sea posible determinar que sensor está midiendo una mayor temperatura. Si ambos voltajes son iguales el voltaje de salida del arreglo será el voltaje de alimentación. El voltaje fluctúa hacia arriba o hacia abajo dependiendo de la lectura de ambos sensores. Utilizando una LUT (*Look Up Table*) es posible entonces determinar el ángulo de alabeo. La diferencia de temperatura medida depende de la ubicación geográfica, factores climáticos como la nubosidad y también de la hora en que se realiza el vuelo. Esta dependencia requiere que el sistema sea calibrado para así poder normalizar las lecturas. La calibración consiste en medir la temperatura del cielo y la tierra (eje Z, que corresponde a la dirección perpendicular a la superficie terrestre). Para obviar la etapa de calibración y hacer un sistema más robusto es posible adicionar un nuevo juego de sensores que constantemente midan la diferencia de temperatura en el eje Z.

3.1.2.3. GPS

El GPS es un sensor que entrega una amplia gama de datos:

- Posición geográfica.
- Altitud sobre el nivel del mar.
- Velocidad de desplazamiento sobre la tierra (*groundspeed*).
- Ubicación de los puntos cardinales y sentido de desplazamiento (rumbo) tanto magnético como verdadero.
- Estado del GPS: si se están o no recibiendo señales de los satélites, número de señales recibidas e intensidad.
- Precisión de la estimación de posición.
- Hora y fecha actual.

En una misión sencilla se le indican al UAV puntos geográficos por los que debe pasar en el transcurso del vuelo. Con los datos entregados por el GPS es posible realizar acciones de control que permitan llegar a los distintos objetivos. Casi todos los GPSs comerciales cuentan con la opción de conectarlos con un computador por medio de USB o puerto serial. Para una aplicación tan específica, como es el caso de este trabajo de Memoria, sólo es necesario conocer los datos más relevantes que brinda la unidad y no contar con los elementos gráficos ni las aplicaciones

de los dispositivos comerciales. Se optó, entonces, por la implementación de un módulo GPS que sólo tuviese interfaz serial para la comunicación con el microcontrolador. En general, el tipo de mensaje que envía el GPS sigue el protocolo NMEA 0183. Este protocolo fue diseñado con el fin de estandarizar los mensajes que envían los distintos instrumentos electrónicos en una embarcación. Detalles del protocolo se pueden encontrar en la página Web de NMEA en [15].

Dentro de las características deseadas del módulo se encuentran:

- Posibilidad de personalizar el mensaje de salida.
- Voltaje de comunicación compatible con el microcontrolador.
- Velocidad de comunicación serial variable.
- Alta frecuencia de refresco de los datos.
- Reducido tamaño.

En Sparkfun [16] es posible encontrar un gran número de módulos GPS. Se optó por elegir un módulo que tuviera integrada la antena y el receptor GPS. Con los requisitos planteados se eligió el LS20031 del fabricante Locosys™ (Figura 3.1.3). Las características destacables de este módulo se muestran en la Tabla 3.1.1.

Tabla 3.1.1: Características del módulo LS20031

Soporta hasta 32 satélites de manera simultánea
Hasta 5HZ de envío de sentencias
Interfaz serial de 57600bps TTL
Voltaje de alimentación de 3.3V
Batería de respaldo incorporada
Precisión de hasta 3m
Personalización de la sentencia enviada

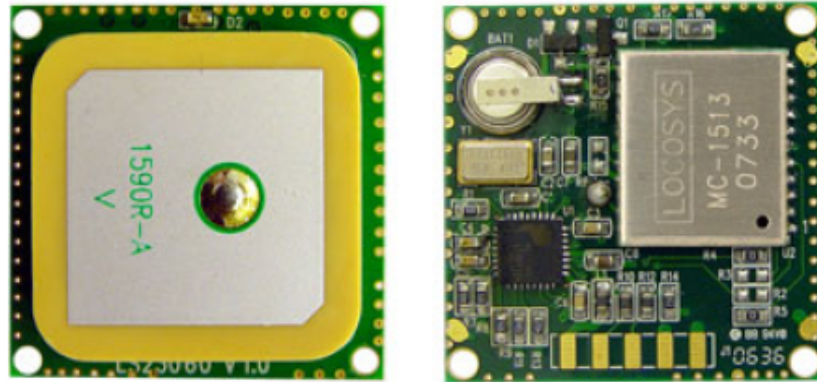


Figura 3.1.3: GPS Locosys™ modelo LS20031

3.1.2.4. Medición de la altura

Para medir la altura se utiliza un sensor de presión absoluta. Dentro de la amplia gama de sensores posibles se eligió uno marca Bosch™, de la línea Bosch™ Sensortec modelo BMP085 (Figura 3.1.4). Los criterios para elegir este sensor fueron:

- Interfaz digital: Entre SPI e I²C se eligió I²C. El protocolo SPI requiere de un puerto dedicado y una salida de control por dispositivo lo que le resta escalabilidad, dado que en el ruteo hay que prever si se van a utilizar dispositivos con esta interfaz de comunicación y dejar las respectivas salidas de control disponibles. En cambio I²C al ser un bus de 2 líneas no requiere de tal planeamiento.
- Reducido tamaño y masa: Todos los modelos son de dimensiones reducidas. Los encapsulados presentan dificultades para el soldado de los integrados. Existen placas de desarrollo (*Breakout*) en los que viene el integrado ya soldado y con sus salidas a pines de fácil uso. La idea es integrar todos los componentes de forma armoniosa en un diseño y es por esto que los desafíos son bienvenidos y no se utiliza el *Breakout*.
- Alta precisión: En el modo de máximo consumo de energía es posible lograr (según la hoja de datos del fabricante en [17]) una precisión en la determinación de la altura de hasta 25[cm].

Este integrado entrega dos mediciones: la temperatura no compensada del sensor y la presión atmosférica no compensada. Con estas 2 mediciones más una serie de parámetros correspondientes

a la calibración interna del integrado es posible realizar una serie de operaciones matemáticas a partir de las cuales se llega finalmente a la temperatura y presión compensada. A partir de la presión es posible determinar la altura sobre el nivel del mar utilizando la siguiente ecuación:

$$Altitud = 44330 * \left(1 - \left(\frac{p}{p_0} \right)^{\frac{1}{5.255}} \right) \quad (3.1.1)$$

Donde p_0 corresponde a la presión atmosférica a nivel del mar y p a la presión medida por el dispositivo. Es interesante destacar que esta forma de determinar la altura a la que se encuentra el dispositivo no determina en realidad la altura con respecto al nivel del mar, si no más bien la presión relativa. Esto se traduce en el hecho de que durante el vuelo, no se considere la altitud real como referencia/parámetro del sistema de control, sino más bien la presión atmosférica. Por lo tanto, la manera adecuada de referirse a este tipo de control no es el del vuelo con referencia de altura con respecto al nivel del mar, es el vuelo por isobaras. La isobara es una *manifold* que une puntos de igual presión atmosférica. En caso de ser necesario obtener la altura real, ésta se puede conseguir de los datos entregados por el GPS. No es conveniente realizar el control de altitud utilizando los datos del GPS, pues estos pueden variar bruscamente dependiendo del error introducido en las lecturas por los dueños y controladores del sistema GPS.



Figura 3.1.4: Fotografía del sensor de presión absoluta

3.1.2.5. Velocidad de vuelo

En un aeroplano la rapidez con la que se desplaza en el espacio corresponde a la rapidez relativa a la tierra. Sin embargo, para efectos del vuelo en sí, esta rapidez no representa una variable de gran utilidad. La rapidez con la que vuela el aeroplano debe ser medida con respecto al medio en el que se desplaza, o sea el aire. Esta rapidez permitirá determinar la sustentación que generan las alas (que es lo que mantiene al avión volando) y de esta forma incrementar o disminuir el empuje generado por el motor. La forma común de realizar esta medición, desde UAVs hasta aviones comerciales de

transporte de pasajeros, es utilizar un tubo de Pitot.

3.1.2.6. Tubo de Pitot

El tubo de Pitot se puede definir como el instrumento para medir velocidades de un flujo mediante la diferencia de presiones estática y total en de un flujo de un líquido. La Figura 3.1.5 muestra un diagrama constructivo del tubo de Pitot.

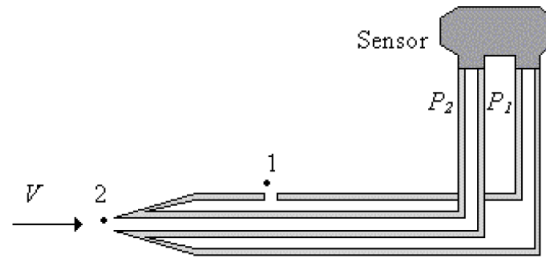


Figura 3.1.5: Tubo de Pitot

El fluido que se mueve a una velocidad V llega al punto 2 en donde se estanca y su velocidad se hace nula. En este punto la energía cinética del fluido se convierte en energía de presión; esta corresponde a la medida de la presión total o de estancamiento (p_t). Del punto 1 se obtiene la medición de la presión estática (p_e). Ambas presiones coinciden en un manómetro diferencial, o sensor de presión diferencial, el cual entrega como resultado la diferencia de las dos presiones: la presión dinámica. La ecuación que relaciona la presión dinámica con la velocidad del fluido se desarrolla a partir de las ecuaciones de Bernoulli obteniéndose como resultado:

$$v = \sqrt{(2/\rho) * (p_t - p_e)} \quad (3.1.2)$$

La medición de la rapidez depende entonces del sensor de presión diferencial. Este tipo de sensor se caracteriza por tener dos entradas y una única salida analógica o digital. La velocidad de vuelo no es una variable extremadamente crítica, no es necesaria una gran precisión en su medición y sólo es necesario mantenerla dentro de ciertos límites. Para disminuir entonces la carga del procesador al momento de realizar la interfaz con el sensor se opta por un sensor con salida analógica. La elección

también se basó en la amplia literatura disponible sobre la utilización de estos sensores en UAVs. El sensor utilizado es fabricado por Freescale™ modelo MPX5010 (Figura 3.1.6).

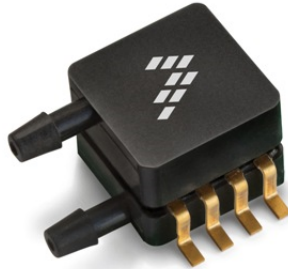


Figura 3.1.6: Sensor de presión diferencial para montaje superficial

3.1.2.7. *Link* RF

Para saber que está realizando el UAV, detectar fallas o ingresar nuevas órdenes, es necesario contar con un medio de comunicación entre el puesto de control (tierra) y el UAV. La forma de implementar esta comunicación es mediante un módulo de radio frecuencia de comunicación bi-direccional (envía y recibe). Como protocolo de comunicación se utilizará el protocolo serial RS-232 dada su fácil implementación tanto en un microcontrolador como en un PC. Módulos de probada fiabilidad son los XBee de la marca Max-Stream™. Estos módulos son bi-direccionales, de variadas potencias de transmisión, pequeño tamaño, fácil implementación y reducido costo. Estos módulos se comportan como un puerto serial transparente tanto para el receptor como el transmisor, no siendo necesario ningún tipo de configuración en los dispositivos transmisor-receptor. El modelo elegido en cuestión es el XBee-PRO 868. Lo atractivo de este módulo por sobre sus competidores es la elevada potencia de transmisión y por lo tanto la gran distancia que es posible cubrir sin perder el enlace. Otra ventaja es la velocidad de transmisión serial que admite. Un detalle de las especificaciones más relevantes se muestra en la Tabla 3.1.2.

Tabla 3.1.2: Características del XBee-PRO 868 [18]

Rendimiento	
Rango en interiores	Hasta 550 metros
Rango en exteriores con línea de vista	Hasta 40 Km
Potencia de transmisión	85-500 mW
Sensibilidad receptor	-112dBm
Interfaz serial	
UART	3.3V CMOS
Velocidad transmisión de datos	1200-230400 bps
Requerimientos de alimentación	
Voltaje de alimentación	3.0 – 3.6 VDC
Corriente operación transmisor	500mA (800mA máx)
Corriente operación receptor	65mA

La frecuencia de transmisión (868Mhz) es una frecuencia con ancho de banda restringido en USA. Por este motivo el módulo tiene un limitador incorporado en el circuito de control de un máximo ciclo de trabajo de 10 %. El inconveniente que esta limitante genera es que al estar transmitiendo, de pronto cese la transmisión. Por este motivo es necesario incorporar un circuito que realice un *reset* del módulo cada cierto tiempo, de modo de saltarse el efecto del limitador. Se adquirió un kit de desarrollo que incluye dos módulos XBee (Figura 3.1.7) y una placa que permite realizar la interfaz XBee-Computador por medio de un puerto USB. Esta placa también permite configurar los módulos para que funcionen con la velocidad y potencia deseada.

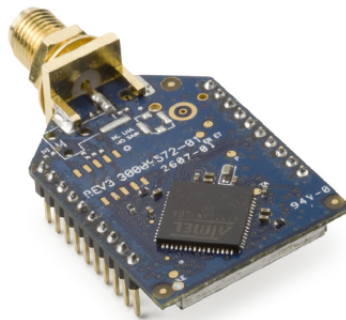


Figura 3.1.7: Ilustración del XBee-PRO 868

3.1.2.8. Elección del microcontrolador

Para las necesidades planteadas y siguiendo con el objetivo de la simplicidad, no se justifica utilizar un PC. La tendencia en este tipo de aplicaciones es la de utilizar DSPs (*Digital Signal Processor*) o MCU (*Micro Controller Units*). Los DSPs, como su nombre lo indica, están orientados al procesamiento digital de señales (audio, video, imágenes, voz) y su estructura está optimizada para cumplir esta tarea. En general necesitan componentes externos para implementar su funcionamiento tales como memorias, conversores de datos y controladores de buses. Los microcontroladores por otra parte son pequeñas computadoras en un mismo integrado; incluyen CPU, memoria y unidades de entrada/salida y están optimizados para realizar tareas de control y comunicación con periféricos. En el UAV no se pretende realizar ninguna tarea que necesite de un DSP, motivo por el cual se buscó dentro de la familia de los microcontroladores uno que se adapte a los requerimientos. La forma de clasificar los microcontroladores depende de sus características, dentro de estas se pueden destacar:

- Memoria: capacidad para almacenar programa.
- RAM: memoria de trabajo.
- Número de pines de entrada/salida.
- Velocidad de procesamiento de la CPU.
- Módulos analógicos: comparadores, conversores análogo-digital.
- Módulos digitales: puertos seriales, SPI, USB, I²C, CAN.
- Temporizadores.
- FPU (unidad de punto flotante).

La Tabla 3.1.3 resume los requisitos de los sensores:

Tabla 3.1.3: Detalle de los puertos de comunicación requeridos

Puertos de comunicaciones requeridos	Detalle
Puerto serial 1	Comunicación con el módulo GPS
Puerto serial 2	Comunicación con el módulo RF
Puerto I2C	Comunicación con el sensor de presión absoluta
Puerto SPI	Posibilidad de utilizar dispositivos que utilicen este protocolo de comunicación
Pines de E/S	
Sensores IR 1 y 2	2 canales ADC
Sensor IR 3 (eje Z no implementado)	1 canal ADC
Airspeed	1 canal ADC
Voltaje alimentación receptor	1 canal ADC
Voltaje alimentación electrónica	1 canal ADC
Expansión analógica	2 canales ADC
Detección conmutación	1 E/S digital
Control de servos	1 PWM
Leds	3 E/S digital
Expansión digital	4 E/S digital

Existe un amplio abanico de microcontroladores que se adaptan a estas necesidades. Sin embargo, la amplia documentación disponible acerca los microcontroladores Texas Instruments™ (TI) y Microchip™ inclinan la balanza hacia estos fabricantes. En experiencias previas se trabajó con microcontroladores PIC de Microchip™, predisponen el uso de éstos por sobre los TI. Se buscó un microcontrolador que contara con la mayor cantidad de periféricos implementados en hardware, a fin de no tener que implementar periféricos por software con el consiguiente uso de procesador. Las restricciones planteadas restringen la gama de microcontroladores a PICs de 16 y 32 bits. Se eligió un microcontrolador de la serie 24F con las siguientes características (Tabla 3.1.4):

Tabla 3.1.4: Características del PIC24FJ128GA006

Arquitectura	16 bits
Velocidad CPU	16 MIPS (cristal 32MHz)
Memoria de programa	128KB
Voltaje de operación	2 - 3.6[V]
Pines E/S	53
Puertos de comunicaciones	2 UART, 2 SPI, 2 I2C
Convertor análogo	16 canales, resolución de 10bits
Velocidad muestreo convertor análogo	Hasta 500.000 muestras por segundo
Temporizadores	5 temporizadores de 16 bits
Módulos de captura y comparación PWM	5 captura y 5 comparación (16 bits)
Interrupciones	43 fuentes de interrupción
Máxima corriente de entrada	250mA
Encapsulado	64 pines TQFP

3.1.2.9. Fuente de alimentación

La fuente de alimentación debe ser diseñada pensando en la fiabilidad dado que cualquier desperfecto en algún componente puede producir un problema mayor y dejar sin control el UAV. Los requisitos energéticos se resumen en la Tabla 3.1.5:

Tabla 3.1.5: Requerimientos eléctricos

Componente	Voltaje alimentación [V]	Consumo (máximo) [mA]
PIC	3.3	300
XBee	3.3	800
GPS	3.3	50
Sensor presión barométrica	3.3	1
Sensor presión diferencial	5	20
Sensores infrarojos	3.3	0

Se observan 3 puntos importantes:

1. Consumo elevado del módulo RF.
2. Voltaje alimentación sensor presión diferencial de 5.0[V].
3. Voltaje alimentación resto de sensores y microcontrolador de 3.3[V].

La alimentación debe provenir de una batería. Actualmente las baterías comercialmente disponibles con la mejor razón densidad de energía/masa son las de litio-polímero (LiPo). Las baterías LiPo se componen de celdas conectadas en serie y/o paralelo dependiendo del fin que se busque lograr, aumento de voltaje y/o aumento de capacidad respectivamente. Existen celdas de

distintas capacidades, pero todas tienen el mismo voltaje nominal 3.7[V]. El voltaje de una celda está por debajo del voltaje de alimentación del sensor de presión diferencial, motivo por el cual la mínima configuración es de 2 celdas en serie (7.4[V]). Los voltajes de alimentación de los distintos componentes electrónicos determinan la necesidad de contar con al menos dos voltaje diferentes: 5.0 y 3.3[V]. La naturaleza del microcontrolador y la utilización de los puertos A/D obligan que su alimentación sea lo más estable posible. El alto consumo de característica pulsante del módulo RF puede hacer que el voltaje en la línea de 3.3[V] baje súbitamente, mermando de esta forma la confiabilidad de algunas características del microcontrolador. Por este motivo se ha tomado la decisión de contar con dos líneas de 3.3[V], la primera para la alimentación del microcontrolador y sensores de bajo consumo y una segunda dedicada a la alimentación del módulo RF. Los reguladores elegidos para cumplir esta tarea son los TLV1117 de Texas Instruments. Las características de este regulador se muestran a continuación en la Tabla 3.1.6.

Tabla 3.1.6: Características del regulador

Voltaje entrada (V_i)	4.7 - 15 [V]
Voltaje salida (V_o)	3.3 [V]
Corriente salida (I)	800 [mA]
Mínimo $V_i - V_o$ para $I = 800\text{mA}$	1.2 [V]
Encapsulado	SOT 223

El sensor de presión diferencial se alimenta con 5[V]. Al ser un sensor analógico la alimentación directa desde la fuente switching no es recomendable. La alimentación correrá entonces por cuenta de un nuevo regulador; el TPS76350 de Texas Instruments [19]. Al igual que el TLV1117, este regulador también es *low dropout*, siendo en este caso más crítica esta característica, pues el voltaje de alimentación es 5.5[V] y el voltaje de salida esperado es de 5.0[V]. Cuando el regulador está entregando máxima corriente (150[mA]) el voltaje de *dropout* es de tan sólo 300[mV], mientras que para 50[mA] es de tan solo 60[mV].

3.1.2.10. Unidad de conmutación y decodificación (conmuta/decod)

En países donde existe una regulación respecto a la utilización de UAVs, como USA, destacan la importancia de contar con una unidad que permita en cualquier momento retomar el control sobre el UAV en caso de ser necesario (*Fail Safe Mode*). El circuito que permite retomar el control del aeroplano es un conmutador. Este conmutador tiene en su salida conectado los servos y en su

entrada las dos señales entre las cuales realizará la conmutación; señal proveniente del receptor y señal proveniente del microcontrolador.

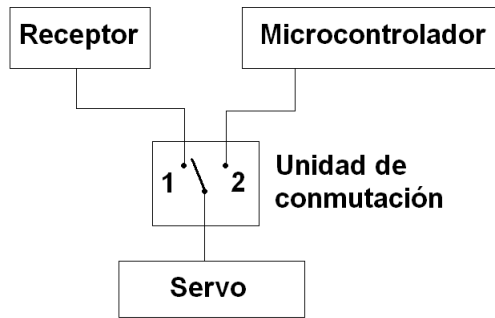


Figura 3.1.8: Esquema de funcionamiento del conmutador

En modo RC (posición 1 del conmutador), los servos reciben las señales provenientes del receptor y en modo autónomo (posición 2 del conmutador) las del microcontrolador (Figura 3.1.8). La señal para realizar la conmutación entre ambos deberá provenir de una fuente segura. Sin embargo, dado que el alcance del radio control es limitado, no tiene sentido devolverle el control al operario si éste se encuentra a una distancia mayor a la que cubre la emisora de radio. Es por este motivo que la señal de control para la conmutación provendrá del equipo RC. La limitación de esta estrategia es que el UAV no podrá realizar misiones en la que la distancia emisora de radio-UAV supere un umbral cercano a los 1.5[km] el cual es ampliamente mayor a la distancia a la cual se pierde de vista un aeromodelo. Esta limitante podrá ser revertida utilizando otro método de control de conmutación, por ejemplo el microcontrolador. Para comprender mejor el funcionamiento de los componentes del conmutador es necesario entender el funcionamiento de los servos.

Servo

El servomotor es un actuador que en su interior tiene un motor de corriente continua (CC), una caja reductora y un circuito electrónico. A diferencia de un motor CC, un servo tiene la capacidad de ubicarse en una posición dentro de su rango de funcionamiento y mantener esta posición. Los servos utilizados en este proyecto son los utilizados en aeromodelismo.



Figura 3.1.9: Servo utilizado en aeromodelismo

En la Figura 3.1.9 se aprecia la placa de circuitos integrados, el motor CC, la caja reductora y el potenciómetro acoplado al eje de salida que hace las veces de codificador de posición. Los servos de aeromodelismo tienen un rango de funcionamiento que varía entre los 90 y 180 grados. Un receptor de aeromodelismo tiene salidas para un número fijo de servos, este número dependerá de la cantidad de funciones que se quieran implementar en el modelo. Cada superficie de mando es movida por un servo. En un aeromodelo básico se necesita un mínimo de 4 servos: alerones, elevador, timón y acelerador. Las salidas del receptor para cada servo se componen de 3 cables: alimentación del servo (V_{cc} y GND, cables rojo y negro de la Figura 3.1.9 respectivamente) y una señal de control (cable amarillo). La electrónica de control del servo recibe esta señal de control y la transforma a una posición angular que es comparada con la posición angular del eje de salida que resulta de la transformación del voltaje entre los terminales del potenciómetro, al igualar estas dos posiciones se logra el control de posición. La técnica utilizada para enviar la señal control es la de modulación de ancho de pulso. La Figura 3.1.10 muestra una señal de control para un servo y la posición angular en el eje de salida que corresponde al pulso de ejemplo.

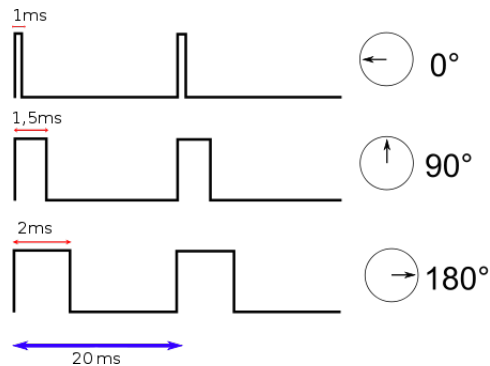


Figura 3.1.10: Ejemplo de señal de control de un motor servo y la posición del eje de salida

La unidad de conmutación interrumpe entonces la señal de control que va desde el receptor al servo. La conmutación entre las señales provenientes del receptor y del microcontrolador es realizada por el integrado CD4053. Este integrado contiene en su interior tres multiplexores de dos entradas cada uno y controlados por tres señales independientes. En el modo de funcionamiento autónomo, el microcontrolador deberá generar las señales de control para los servos; el número de señales generadas depende del número de servos que se desee controlar. Esta forma de enviar las señales de control obliga la asignación de un número de “pines” del microcontrolador igual al número de servos a controlar, aumentando la cantidad de señales a rutear en la etapa de implementación y perdiendo funcionalidades del microcontrolador. Basado en el funcionamiento de un receptor se encontró una solución que requiere de sólo un “pin” del microcontrolador para manejar hasta 9 servos. La conversión de una señal de control en hasta 9 señales para los servos es realizada en un receptor por un integrado denominado “Frame-Decoder”. El Frame-Decoder no es más que un contador de señales de reloj. Con cada flanco de subida de la señal de entrada “clock” al integrado el contador aumenta en una unidad y la salida correspondiente al estado del contador se “enciende”. El diagrama (Figura 3.1.11) de señales ejemplifica el funcionamiento del integrado CD4017.

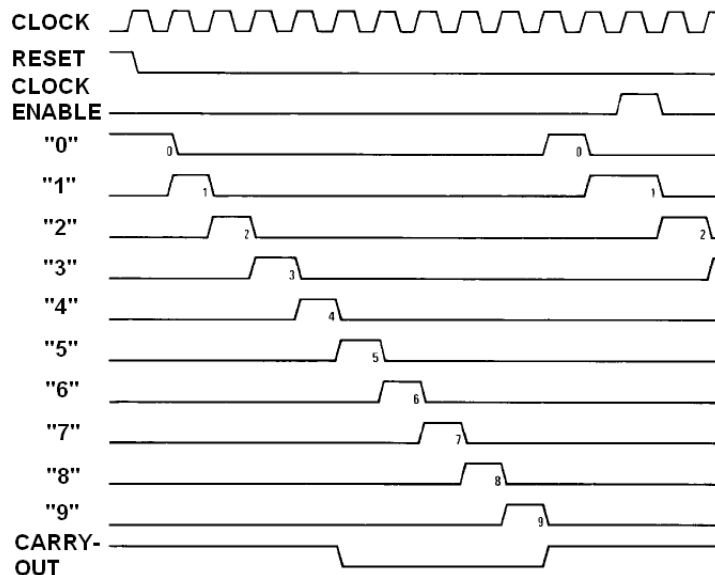


Figura 3.1.11: Diagrama de sincronía del CD4017 [20]

Cada salida está conectada a un servo y la señal de reloj (*clock*) proviene del microcontrolador. De esta forma, administrando la señal del reloj es posible generar el PWM para cada uno o de los

servos como se puede ver en el ejemplo de la Figura 3.1.12:

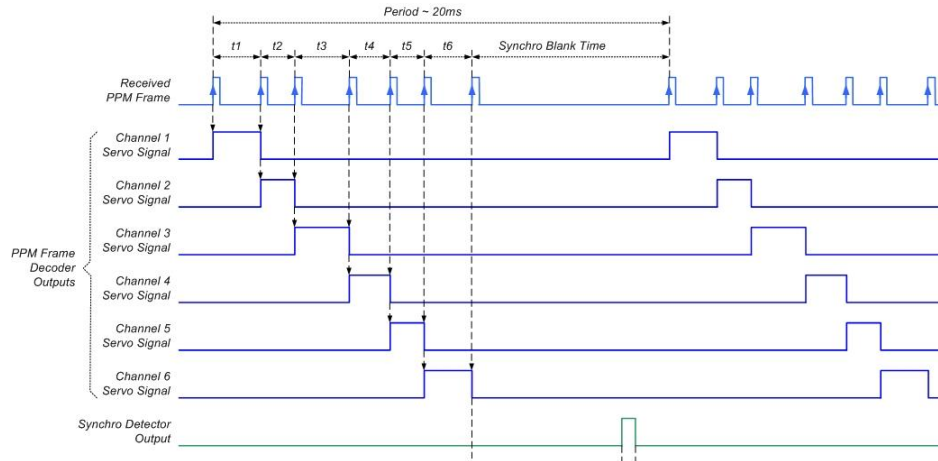


Figura 3.1.12: Funcionamiento frame decoder [21]

Como la señal para realizar la conmutación proviene del equipo RC, es necesario contar con un dispositivo que se conecte como un servo más al receptor y que utilizando las mismas señales de control de un servo sea capaz de identificar cuando se desea realizar la conmutación. Esta tarea la puede realizar sin ningún problema un microcontrolador de la gama más baja, como un PIC12F509. La radio cuenta con interruptores que se pueden utilizar para controlar un servo que adopte 2 ó 3 posiciones dependiendo de la posición del interruptor. Estos canales son generalmente utilizados por ejemplo para accionar la subida o bajada el tren de aterrizaje de un modelo que posea esta capacidad. El microcontrolador dedicado a detectar la conmutación tiene que simplemente medir, utilizando una de sus entradas, el ancho del pulso proveniente del receptor y activar una de sus salidas si la duración del pulso supera un umbral. Se prefirió utilizar un microcontrolador por sobre electrónica digital (Flip-Flops) debido a las interferencias en el receptor que podrían gatillar inesperadamente la conmutación de modo RC a modo autónomo. En el microcontrolador es posible programar una rutina que filtre estas interferencias y que en el futuro reciba cualquier tipo de señal para realizar la conmutación, por ejemplo del microcontrolador del piloto automático.

Un esquema de toda la electrónica y su interconexión se muestra en la Figura 3.1.13.

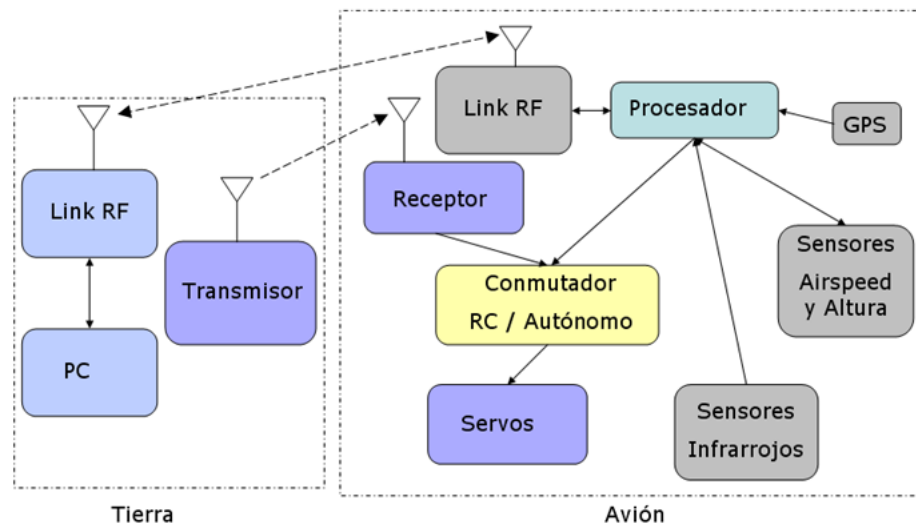


Figura 3.1.13: Diagrama de bloques del sistema completo

3.1.2.11. *Link* serial-USB

En la actualidad es muy difícil encontrar un computador portátil, y cada vez menos de escritorio, que cuenten con un puerto serial y en vista que éste es el protocolo más abundante en el sistema de control, se implementará un convertor de serial a USB. Existen integrados dedicados a esta conversión y uno de los más utilizados es el FTDI232. Al instalar el controlador de este integrado en un computador se obtiene un puerto serial con el cual es fácil realizar una interfaz con programas como MATLAB®.

3.2. Diseño del aeroplano

El aeroplano se debe diseñar teniendo en mente que es una plataforma para el transporte de sensores que además debe cumplir con requisitos de maniobrabilidad y de robustez como fue descrito en los objetivos. Las características del aeroplano deben ser entonces las siguientes:

- Gran estabilidad.
- No requerir gran experiencia para ser piloteado.
- Capacidad de transportar la electrónica necesaria para la automatización.
- Espacio disponible para montar más electrónica en trabajos futuros.

- Gran autonomía.
- Construcción sencilla utilizando materiales modernos.
- Fácil reparación en caso de siniestro.

3.2.1. Solución propuesta para el diseño del aeroplano

Existe un viejo dicho en la aviación: *“si se ve bien, volará bien”*. Si bien este dicho se cumple en la mayoría de los casos, no implica que haya que dejar de lado los cálculos. La estética se acomoda de tal forma de lograr un avión que se vea bien y esté bien calculado.

3.2.1.1. Elección del tipo de configuración

La estabilidad, facilidad de volar y autonomía llevan a elegir un aeroplano de tipo “ala alta” y gran envergadura (aprox. 3[m]). La electrónica para la automatización no sugiere especial cuidado en el diseño pues es de reducidas dimensiones en comparación con el aeroplano. Sin embargo, en caso de querer montar más sensores en el futuro, es necesario considerar el espacio necesario y la forma de tener acceso a la mayor porción posible del fuselaje para realizar la instalación. La configuración más común de un aeroplano es la mostrada en la Figura 3.2.1 (izquierda). Este tipo de configuración cumple con los requisitos de estabilidad y dificultad de pilotaje. Sin embargo al momento de evaluar la incorporación de nuevos sensores aparece el problema del motor y la hélice, por ejemplo en el caso de querer montar una cámara apuntando en la dirección de vuelo. Esta limitante se soluciona diseñando el aeroplano con una configuración “pusher” (motor montado con la hélice-motor en la parte posterior, en lugar de la anterior). La configuración pusher tiene otras ventajas como la maniobrabilidad a bajas velocidades (despegue y aterrizaje) y el amplio espacio en la parte anterior del aeroplano separado del motor y estanque de combustible. Este es el motivo por el cual la mayoría de los UAVs ocupan una configuración pusher, como el Aerostar (derecha Figura 3.2.1). Un fuselaje tipo “cajón” brinda un amplio espacio interior a la vez de ser muy resistente.



Figura 3.2.1: Configuración tractora y pusher [22]

3.2.1.2. Diseño del aeroplano

En una primera instancia se definen dimensiones gruesas del aeroplano que se quiere construir. Lo primero a definir es el ala y su forma; ala rectangular de una envergadura no superior a 3 metros. La carga alar es el punto de partida para todos los siguientes cálculos. Se eligió un valor promedio de cargas alares recomendadas: $75 \left[\frac{gr}{dm^2} \right]$. Este valor de carga alar, si bien no es bajo, ayuda a la estabilidad del avión. Con la carga alar definida es necesario determinar la masa aproximada del aeroplano. En aviones de similares dimensiones y no acrobáticos los motores típicamente utilizados son de 30[cc]. El aeroplano contará con alerones, timón de profundidad, timón de dirección y acelerador. Con la configuración pusher y las dimensiones estimadas son necesarios 7 servos:

- 2 para los alerones, uno en cada semi-ala.
- 2 para cada uno de los timones de dirección.
- 1 para el elevador.
- 1 para el acelerador.
- 1 para la dirección de la rueda delantera que brinda control del avión en tierra.

Se procede a confeccionar la Tabla 3.2.1 con los valores aproximados de las masas:

Tabla 3.2.1: Valores aproximados de las masas

	Masa[gr]	Masa[oz]
Motor	1200	
Mezcla	800	
Receptor	40	
Servos	500	
LiPos	400	
Tren At.	300	
Alas	2000	
Fuselaje	1000	
Cola	1000	
Electronica UAV	600	
Total =	7840	279.888

A partir de la carga alar y la masa esperada es posible determinar la superficie alar:

$$Sup = \frac{masa}{carga_{alar}} \quad (3.2.1)$$

Tabla 3.2.2: Superficie alar

	dm^2	f^2
Sup. Alar :	104.5333	11.19552

Una variable importante en la forma de un ala es el “alargamiento” (parámetro adimensional definido por la ecuación $alargamiento = \frac{envergadura}{cuerda}$, donde *cuerda* es la distancia entre el borde de ataque y fuga del ala). A mayor alargamiento menor será la resistencia. Los planeadores usualmente tienen alas con un alargamiento entre 10 y 15 y los aviones acrobáticos entre 4 y 6. Para un aeroplano que tiene una velocidad de vuelo moderada un alargamiento de 7 es adecuado. Definido el alargamiento y la superficie alar es posible determinar la envergadura y la cuerda del ala:

Tabla 3.2.3: Envergadura y cuerda del ala

	<i>cm</i>
Envergadura :	270.5057
Cuerda :	38.64367

La forma de los planos de cola está relacionada tanto con su posición como con el ala en sí. El brazo de palanca fija la distancia entre el primer tercio de la cuerda media del ala y el primer tercio

de la cuerda media del estabilizador horizontal. Un valor razonable para el brazo de palanca esta comprendido entre 2 y 3 veces la cuerda media del ala, se elige un valor medio de 2.5. La superficie del estabilizador viene dado por la siguiente fórmula:

$$Sup.E.H = \frac{0,52 * SupAlar * 100 * Cuerda}{Brazopalanca * cuerda} \quad (3.2.2)$$

Tabla 3.2.4: Superficie estabilizador horizontal

	<i>cm</i> ²
Sup. Estab H :	2174.293

El alargamiento del estabilizador debe ser menor al alargamiento del ala y un valor entre 2.5 y 5 es aceptable. Al fijar la envergadura se obtiene la cuerda y el alargamiento. Se fija una envergadura que de cómo resultado un alargamiento cercano a 4:

Tabla 3.2.5: Envergadura, cuerda y alargamiento estabilizador horizontal

	<i>cm</i>		
Envergadura Estab. H :	95		
Cuerda Estab :	22.8873	Alargamiento Estab. H :	4.150774

La cuerda de los estabilizadores verticales es la misma del estabilizador horizontal. A partir de la siguiente fórmula se obtiene el área que deben tener los estabilizadores verticales en función de la superficie del estabilizador horizontal.

$$SupE.V = (0,8 * \frac{SupE.H.}{2}) \quad (3.2.3)$$

Y a partir de la superficie y cuerda es fácil determinar la envergadura:

Tabla 3.2.6: Superficie, cuerda y envergadura estabilizador vertical

	<i>cm²</i>
Sup Estab. V :	869.7173
	<i>cm</i>
Cuerda :	22.8873
Envergadura :	38

Las superficies de mando (alergones, elevador y timones) se diseñan en base a porcentajes de las superficies totales a las que pertenezcan:

Tabla 3.2.7: Superficie de las superficies de mando

Alerones	
% superficie alar	7.5
Superficie total [<i>cm²</i>]	783.975
Envergadura [<i>cm</i>] semiala	60
Cuerda [<i>cm</i>]	6.533125
Elevador	
Envergadura Elevador [<i>cm</i>]	66
% superficie E.H. elevador	27
Cuerda elevador [<i>cm</i>]	8.894836
Timón	
Envergadura Timón [<i>cm</i>]	38
% superficie E.V. timon	28
Cuerda timon [<i>cm</i>]	6.408444

3.2.1.3. Elección del perfil alar

La elección del perfil alar determinará gran parte de las cualidades del avión en vuelo. Se buscaron perfiles de aeromodelos de entrenamiento de envergaduras similares al avión diseñado. Se

identificaron 3 perfiles como los más utilizados:

- Naca 4415.
- Clark Y.
- Gottingen 398.

Las curvas polares definen las características de un perfil alar. La curvas polares de los perfiles propuestos se muestran en la Figura 3.2.2:

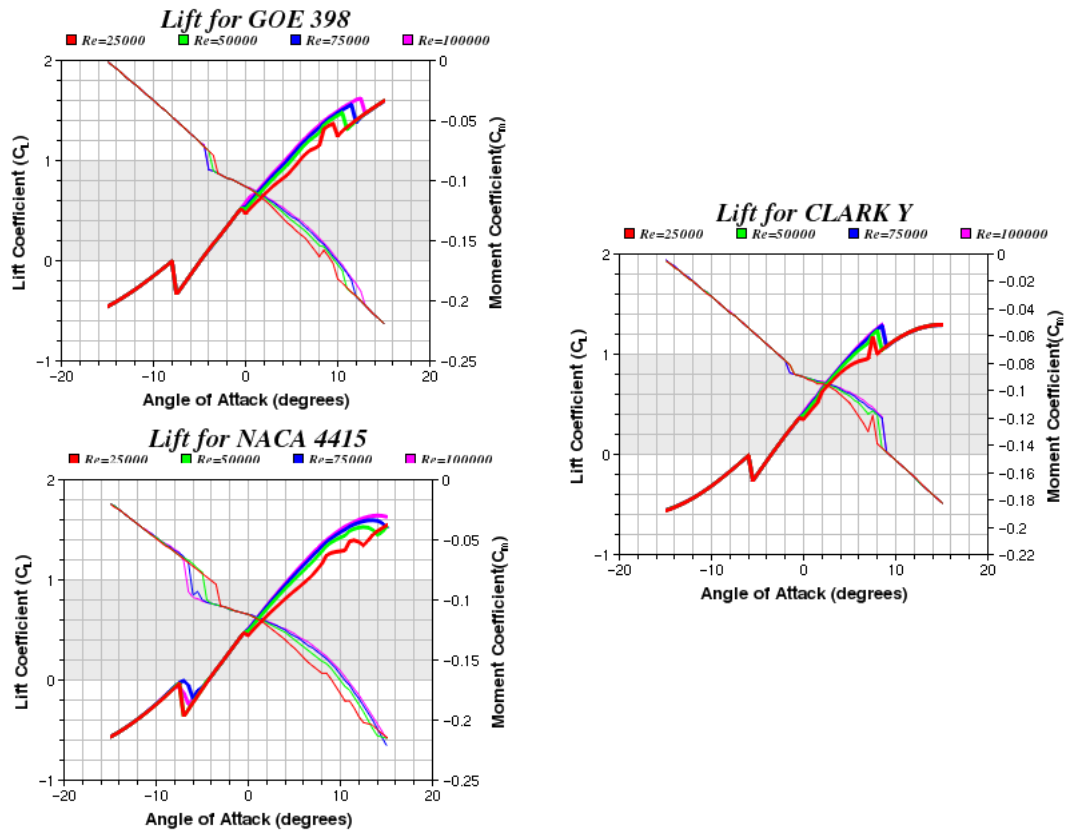


Figura 3.2.2: Curvas polares de los perfiles alares estudiados [23]

De las curvas polares y de las características geométricas de los distintos perfiles es posible generar la siguiente tabla comparativa (Tabla 3.2.8):

Tabla 3.2.8: Tabla comparativa de perfiles

	GOE 398	NACA 4415	CLARK Y
Espesor [%]	13.8	15	11.7
Camber [%]	4.9	4	3.4
Max Cl (coeficiente de sustentación)	1.622	1.643	1.295
Max ángulo Cl	12.5	14	8.5
Angulo Zero-Lift	-4.5	-4	-3.5

Un perfil de espesor grueso como el NACA o el GOE, desde el punto de vista constructivo, favorecen un ala más rígida. El *camber* indica lo “flotador” del perfil. El mayor % de *camber* incrementa el Cl max y desplaza hacia la izquierda el ángulo de Zero-Lift, lo que se traduce en un vuelo más estable. Dada las características esperadas tanto de vuelo como constructivas, se optó por el GOE 398. El bajo espesor del Clark y el muy grueso del Naca implican baja rigidez y elevada masa, respectivamente.

3.2.1.4. Elección de la planta motriz

Una vez que se ha definido la cilindrada, sólo falta determinar el fabricante de éste. Los precios de los motores de este tamaño de combustión interna, bencineros, pueden fluctuar entre los USD\$150 hasta los USD\$800. Los motores de origen Chino del fabricante DLE han probado ser confiables, livianos, poderosos y de un precio muy asequible. Otra ventaja por sobre otras marcas es la posibilidad de encontrar los repuestos en Chile. El modelo elegido es el DLE 30 y la Tabla 3.2.9 muestra sus principales características:

Tabla 3.2.9: Características motor DLE 30

Rendimiento	3.7 HP / 8500 RPM
Relantí	1600 RPM
Empuje	8.5 Kg / 100metros de altura
Cilindrada	30.5 cc
Masa motor	1100 gr
Masa electrónica	60 gr
Hélices recomendadas	18x8, 18x10, 19x8, 20x8

3.2.2. Estrategia de control

Este trabajo busca dejar una plataforma sobre la cual poder seguir desarrollando nuevos controladores e incorporar nuevos sensores, motivo por el cual el control a implementar no será complejo. Las variables controladas son los ángulos de alabeo y cabeceo y las variables manipuladas la posición angular del eje de salida de los servos de los alerones y elevador. Los ángulos de cabeceo y alabeo se determinan utilizando los sensores infrarrojos. Se implementan dos controladores independientes proporcionales para cada uno de los ángulos a controlar. El controlador será definido por tramos, para contar con ganancias dependientes de la magnitud del error. Al igual que en los trabajos revisados, la modelación de un aeroplano para probar y sintonizar controladores resulta una tarea muy compleja, lo que lleva a realizar una sintonización en el campo de vuelo utilizando algoritmos de prueba y error. La disposición de los sensores infrarrojos en el aeroplano se muestra en la Figura 3.2.3.

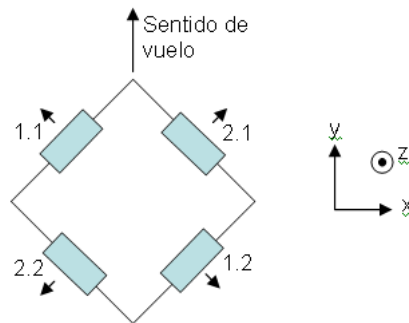


Figura 3.2.3: Disposición de los sensores infrarrojos

Se eligió esta configuración de sensores montados en 45° con respecto a la dirección del vuelo por sobre la orientación 0 y 90° dada la literatura que sugiere una mayor precisión con la montura en 45° ya que no existe obstáculos visuales como el fuselaje o las alas [24]. La lectura obtenida corresponde a la diferencia de voltaje de un par de sensores (1.1-1.2 y 2.1-2.2). Hay que recordar que es necesario realizar una calibración previa al vuelo para determinar las diferencias máximas en las lecturas de los sensores: esto implica rotar el aeroplano en torno al eje X e Y.

La Figura 3.2.4 muestra como varía la temperatura con respecto al ángulo de rotación.

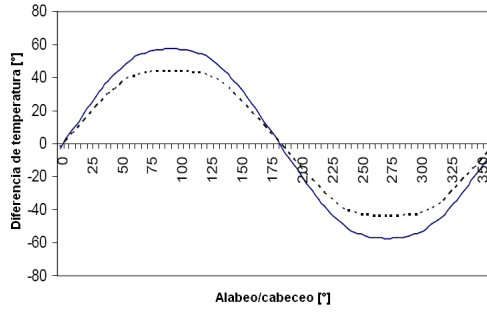


Figura 3.2.4: Variación temperatura con respecto a ángulo de rotación [25]

Para determinar la actitud del aeroplano es necesario realizar una combinación de ambos pares de sensores.

$$Lectura_{cabeceo} = (adc_2 - neutraladc_2) - (adc_1 - neutraladc_1) \quad (3.2.4)$$

$$Lectura_{alabeo} = (adc_2 - neutraladc_2) + (adc_1 - neutraladc_1) \quad (3.2.5)$$

Donde:

adc_x = corresponde al valor obtenido al realizar la conversión análogo-digital del canal x .

$neutraladc_x$ = corresponde a la lectura cuando el aeroplano se encuentra estabilizado.

Con el parámetro de calibración es posible construir una senoide cuya amplitud sean los máximos registrados en la fase de calibración. Ahora simplemente resta encontrar la correspondencia lectura actual – ángulo alabeo/cabeceo. Esto se puede realizar utilizando la función inversa del seno o una LUT normalizada con los parámetros de calibración.

Una vez determinados los ángulos es posible realizar una acción de control que permita estabilizar al aeroplano tanto en el cabeceo como en el alabeo. Esta acción de control se realiza sobre los servos que mueven las superficies de mando. Para controlar los servos es necesario transformar la salida del controlador en un ancho de pulso, que es la forma de enviarle información al servo. Esta etapa se realiza implementando una LUT que transforme la entrada en un ancho de pulso.

Un esquema del sistema de control que será implementado en el microcontrolador se muestra a continuación en la Figura 3.2.5.

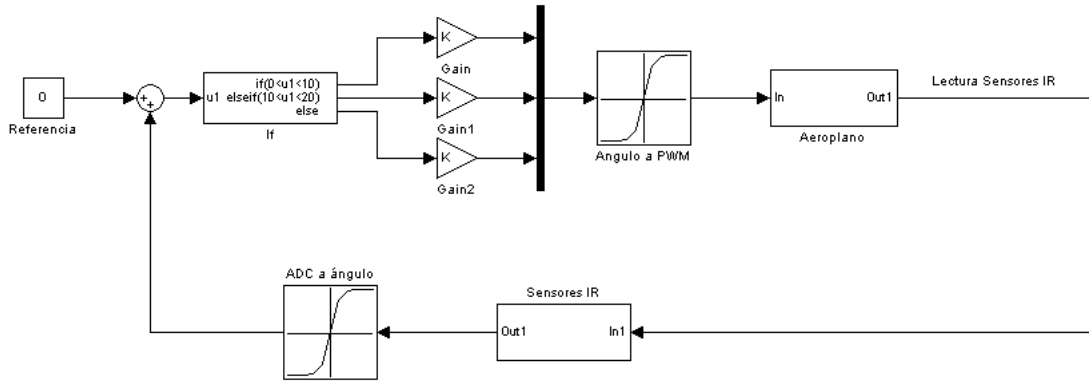


Figura 3.2.5: Esquema del sistema de control

Capítulo 4

Implementación y construcción del prototipo

En este capítulo se presenta el detalle técnico de la implementación de las etapas descritas en el Capítulo 3. Los programas y herramientas utilizadas se especifican a lo largo del capítulo. También se analizan las dificultades encontradas en el proceso y como fueron solucionadas.

4.1. Electrónica

Con los componentes ya elegidos se procede a realizar el diseño y construcción de la placa de circuitos impresos (PCB). El método utilizado para la confección del PCB elegido es el de prototipado rápido. El Departamento de Ingeniería Eléctrica de la Facultad cuenta con una máquina de procedencia alemana, marca LPKF™ y modelo ProtomatC60 (de ahora en adelante simplemente LPKF), construida específicamente para la producción de PCBs. Como se han elegido componentes de pequeñas dimensiones, los métodos basados en corrosión del cobre no son suficientes para lograr la precisión deseada. Además se presentan las pruebas realizadas a la electrónica que aseguran su correcto funcionamiento.

4.1.1. Programas utilizados

El diseño y construcción de un PCB se divide en una serie de etapas. Cada una de estas etapas tiene asociado un software especializado. Los archivos generados por cada uno de estos softwares son la base para que los softwares de las siguientes etapas puedan ejecutar sus respectivas funciones. Los programas utilizados y su función se detallan a continuación:

Diseño y ruteo

EAGLE® 5.4.0 (desarrollado por CadSoft™)

EAGLE®, acrónimo de Easily Applicable Graphical Layout Editor, es un programa que permite el diseño del esquemático así como también el ruteo. El diseño del esquemático es el primer paso en la creación de un PCB. En esta primera instancia sólo se toman en cuenta las funcionalidades de los componentes y la interconexión mediante señales entre los distintos componentes representados por símbolos, no es necesario conocer su forma ni distribución física. Es en el ruteo, cuando el mundo abstracto lleno símbolos del esquemático se transforma al mundo real, donde se trabaja con la disposición de los componentes y las señales en forma de pistas sobre la placa.

El proceso del ruteo genera los archivos Gerber que son utilizados por la siguiente etapa.

Construcción

CircuitCAM® 5.0 (desarrollado por LPKF™)

Es el primero de dos softwares dedicados de la LPKF. La materia prima sobre la cual actúan la herramientas (fresas y taladros) es una placa de dieléctrico cubierta de cobre sobre el cual se dibujarán las pistas y los pads de los componentes. Para lograr esto, es necesario que la máquina sepa donde debe fresar y taladrar y así remover el cobre necesario para aislar las pistas unas de otras. CircuitCAM® es el programa que transforma los archivos Gerber del ruteo en órdenes para las herramientas. En esta etapa se definen también las distintas capas a trabajar, los procesos para las distintas herramientas y los límites de la placa para la herramienta que realiza el dimensionado del PCB.

BoardMaster® 5.0 (desarrollado por LPKF™)

Es la interfaz final usuario-máquina. Los archivos generados por el CircuitCAM® son interpretados por el BoardMaster® quién controla lo motores de la fresadora para realizar las operaciones deseadas. En este programa se monitorea el avance del trabajo en tiempo real y el usuario es informado cuando debe cambiar las herramientas para comenzar una nueva etapa de fresado. Esto es para el caso de la máquina con que se cuenta en el Departamento. En versiones industrializadas los cambios de herramienta se hacen de manera automática. La utilización de la LPKF necesita de una persona capacitada, pues cualquier error de calibración o postura de las herramientas deriva en una herramienta rota o una falla mayor.

Programación

PICC® 4.084 (desarrollado por Custom Computer Services™)

Es el programa utilizado para escribir y compilar el código a grabar en el microcontrolador. El lenguaje de programación es C orientado a los microcontroladores de la marca Microchip™. El compilador incluye una gran cantidad de librerías que facilitan la tarea de programación como por ejemplo: lectura de valores de los conversores ADC, manejo de “pines” E/S, implementación y manejo de puerto serie y bus I²C por hardware, temporizadores y manejo de interrupciones. Otra ventaja del programa es que cuenta con un “*wizard*”, en el cual se puede realizar fácil e interactivamente una configuración inicial del microcontrolador y de las funciones a utilizar. Este programa genera los archivos de salida necesarios para grabar el PIC. Una herramienta muy práctica, incluida en el programa, es el SIOW® (monitor del puerto serial). Con este programa es posible leer y escribir en el puerto serial y de esta forma comunicarse con el microcontrolador.

PICkit 2 Programmer 2.61.00 (desarrollado por Microchip™)

El PICkit 2, fabricado por Microchip™, es un programador y un depurador (*debugger*) que se conecta al puerto USB del computador. El kit se divide en una parte software y otra hardware. En el software se carga el archivo .hex (generador por el PICC® u otro compilador) que será grabado en el microcontrolador y el hardware es la interfaz entre el microcontrolador y el computador. Con el programador también se puede verificar el estado del PIC y determinar si se ha grabado correctamente o si está dañado.

4.1.2. Esquemáticos

Los sensores y el microcontrolador forman la base entorno a la cual se arma todo el circuito. Cada uno de los componentes principales; sensores, fuente de alimentación y microcontrolador, requieren de una serie de componentes externos como; resistencias, condensadores, amplificadores operacionales, inductancias y cristales para su correcto funcionamiento. En la hoja de datos (*datasheet*) de cada componente es posible encontrar las configuraciones más comunes para su utilización y se especifican fórmulas matemáticas para el cálculo de los componentes adicionales a utilizar. En base a estas configuraciones se diseñaron los esquemáticos de los distintos bloques que componen el circuito:

- Sensores IR.
- Fuente de alimentación.
- Sensores de presión absoluta y diferencial.
- Conmutador modo RC-autónomo y decodificador señal servos.
- Módulo de comunicación mediante RF.
- GPS.
- Microcontrolador.

Si bien la librería de componentes disponibles en el EAGLE® es amplia, existen componentes que son muy específicos o muy recientes y por lo tanto no se encuentran incluidos. En el *datasheet* de cada componente es posible encontrar un diagrama funcional, utilizado en el esquemático, y la descripción física del encapsulado, utilizado en el ruteo. Utilizando estos datos y la herramienta para crear nuevas librerías con que cuenta EAGLE®, es posible utilizar los antes inexistentes componentes en la librería original en nuevos diseños.

Dada la condición de prototipo y las probabilidades de que alguno de los componentes deba ser reemplazado se decidió implementar todo el circuito de forma modular. Los módulos van montados sobre una placa madre utilizando *pinheaders* y *sockets* que permiten el fácil reemplazo de estos. Otra ventaja que brinda la implementación modular es la posibilidad de realizar pruebas a los módulos de forma independiente, lo que facilita el debugueo de los distintos programas y puesta en marcha de los periféricos.

Dentro de los requerimientos se encuentra la posibilidad de agregar nuevos sensores o periféricos al sistema implementado. Con el fin de cumplir con estos requerimientos se han dejado líneas de alimentación en distintos voltajes, 1 puerto SPI, bus I²C, puertos de captura/PWM, pines de interrupción externa del microcontrolador y pines que corresponden a canales ADC o GPIOs si se requiere.

Módulo Alimentación

El módulo de alimentación es el encargado de alimentar con energía eléctrica casi todos los componentes del sistema. El sensor de presión diferencial tiene un muy bajo consumo y requiere una tensión de alimentación de 5.0[V]. Dado que ningún otro componente se alimenta con esta tensión, se optó por incluir en el módulo del sensor un regulador que utiliza la línea regulada de 5.5[V] para obtener los 5.0[V]. El bajo consumo asociado al regulador #3 permite dejar esta línea de 3.3[V] abierta a usos posteriores que requieran esta tensión de alimentación. Lo mismo sucede con la fuente #1, a esta línea se pueden conectar más reguladores para alimentar dispositivos que tengan otros requerimientos de voltaje o consumo. El diagrama de bloques de la Figura 4.1.1 que describe la alimentación del sistema (los componentes que se encuentran en el módulo de alimentación están encerrados por el recuadro punteado) se muestran a continuación:

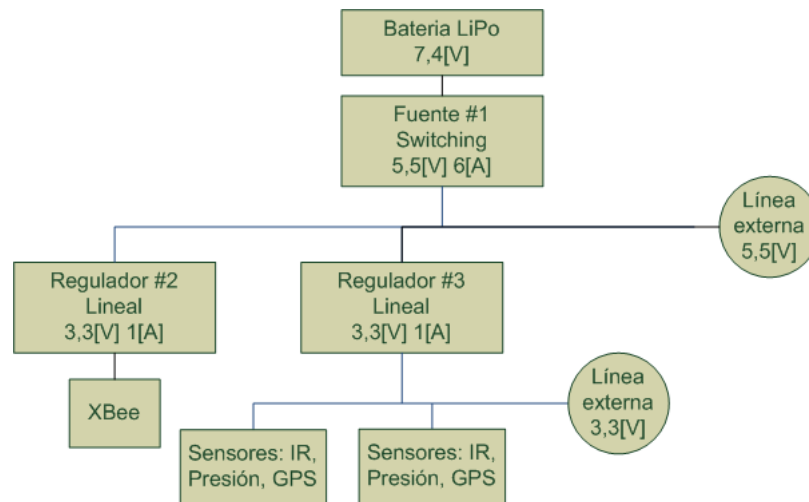


Figura 4.1.1: Diagrama de bloques del módulo de alimentación

La ventaja que posee esta fuente switching, es que sólo requiere de un componente externo para su funcionamiento. Según la fórmula presentada en el *datasheet* [26]:

$$R_{set} = \frac{1,182}{V_o} - 0,591 [K\Omega] \quad (4.1.1)$$

La resistencia requerida para lograr esta tensión (5.5[V]) es de 241[Ω]. Como filtro adicional se colocaron condensadores electrolíticos de 100[uF] en la entrada y la salida. Los reguladores lineales no requieren de componentes externos tampoco, pero el fabricante recomienda el uso de condensadores, electrolíticos o de poliéster, de 10[uF] y 100[uF] a modo de filtro en la entrada y salida respectivamente como se indica en el *datasheet* [27].

El sistema comprendido por el microcontrolador, sensores y puertos de expansión queda definido en el siguiente diagrama de bloques, en donde lo encerrado por la línea punteada comprende la placa madre (Figura 4.1.2):

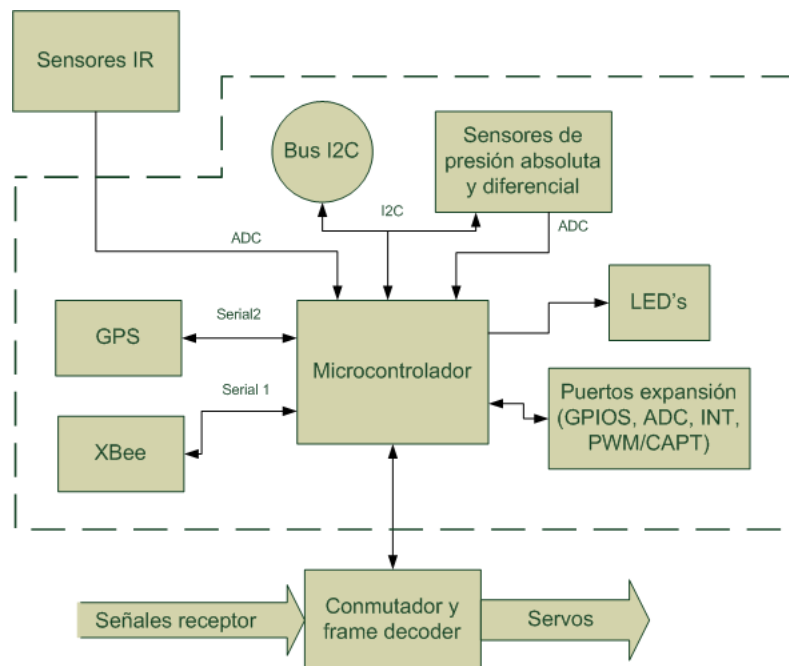


Figura 4.1.2: Diagrama interconexión placa madre

Módulo Conmuta/decod

La alimentación del detector de conmutación y el *frame decoder* es la misma que la alimentación de los servos y receptor de radio. Esta alimentación proviene de una LiPo distinta a la alimentación

de la electrónica de automatización y cuenta con su propio regulador. La estrategia de contar con diferentes fuentes de alimentación se basa en la probabilidad que ocurra algún tipo de desperfecto en el circuito de la placa madre que produzca una falla y una consecuente pérdida de alimentación en los servos y el receptor de radio.

El microcontrolador y el módulo conmuta/decod están unidos por sólo 3 señales referenciadas a tierra; señal que le indica al microcontrolador si se ha realizado una conmutación, señal PWM que genera el microcontrolador para el *frame decoder* y finalmente la señal de reset para el *frame decoder*. Al ser sólo 3 señales las que unen la placa madre y el módulo conmuta/decod se decidió dejar este módulo independiente del resto.

Dado que las señales de los servos son básicamente pulsos, se añadieron condensadores de desacople en la alimentación de los integrados multiplexores y en el *frame decoder*. En las entradas de las señales de los servos a los multiplexores se añadieron resistencias en configuración *pull-down* (conectadas a tierra). Estas resistencias impiden las reflexiones en la línea de señal del servo: si no hay carga en la línea la señal puede rebotar (dada la elevada impedancia de entrada de las entradas del integrado) causando efectos no deseados como interferencia. La línea PWM y la de reset provenientes del microcontrolador, que entran al frame decoder, también tienen resistencias *pull-down*. En el supuesto que el microcontrolador sufra algún desperfecto y sus salidas queden en alta impedancia, las resistencias *pull-down* obligarán a estas señales a permanecer a tierra. Al igual que las resistencias de entrada de las señales del servo, su uso en este caso también ayuda a evitar reflexiones.

En la configuración que se utilizará el PIC12F509 no se necesitan componentes externos para su funcionamiento. Cuenta con un oscilador interno de 4[MHz] que es suficiente para realizar la medición del ancho del pulso enviado por el receptor en canal destinado a la conmutación. Por regla general, en la alimentación del PIC se incluyó un condensador de desacople. Se añadió también una resistencia *pull-down* en la señal de entrada y resistencias en serie, limitando corriente, con las señales salientes del PIC. Siempre es de utilidad contar con un dispositivo de salida monitoreable sin herramientas especiales, un LED conectado a una salida disponible en el PIC cumple esta función. Agregar LEDs en las salidas no utilizadas es una “deformación” adquirida por la práctica, la experiencia ha demostrado que son una herramienta de observación de estados y debugueo infalible.

Las librerías de los integrados no se encuentran en la librería incluida en EAGLE® por lo que fue necesario crearlas.

Módulo sensores de presión

En un mismo módulo se incluyen los dos sensores de presión: sensor de presión absoluta y sensor de presión diferencial.

El sensor de presión absoluta requiere de sólo un condensador cerámico de 0.1[uF] en la alimentación. La comunicación con el microcontrolador es mediante el protocolo I²C. Por definición el protocolo I²C requiere de resistencias *pull-up* en ambas líneas de transmisión; datos (SDA) y reloj (CLK). Estas resistencias son las que le brindan la flexibilidad al protocolo a la hora de conectar dispositivos alimentados con distintas tensiones. Los puertos I²C están diseñados como colectores abiertos, o sea, se comportan como un interruptor a tierra cuando están cerrados y cuando están abiertos la tensión en la línea será la tensión de las resistencias *pull-up*. Como el bus I²C quedará abierto a otros dispositivos en forma de *pinheader* en la placa madre, las resistencias *pull-up* no irán en el módulo de presión, sino que en la placa madre.

Como se mencionó en la anterior descripción del módulo de alimentación, la alimentación del sensor de presión diferencial es proveída por un regulador incorporado en el módulo de presión. La configuración utilizada es la propuesta por el fabricante en el datasheet [28]. La salida del sensor es un voltaje que llega a un canal ADC del microcontrolador. Para evitar atenuaciones en esta línea y no interferir en la medición se agregó un amplificador operacional (*opamp*) en configuración de seguidor de voltaje. El *opamp* elegido es el TLV2472 debido a su característica *rail-to-rail*, lo que implica que el voltaje en la salida puede estar comprendido en todo el rango de alimentación del *opamp*. La salida del sensor tiene un rango también similar a la alimentación, por lo que un *opamp* sin la característica *rail-to-rail* no sirve, ya que la salida se saturaría con voltajes cercanos a 0[V] y 5[V]. Al estar la salida del sensor comprendida en todo su rango de alimentación (0 - 5[V]) no es posible conectarla directamente al microcontrolador, pues los canales ADC aceptan sólo voltajes comprendidos dentro del rango de alimentación del microcontrolador (0 - 3.3[V]). Utilizando un divisor de tensión se logra escalar la lectura a un rango seguro para el microcontrolador. Este divisor se incorporó al módulo de presión.

La naturaleza análoga del sensor de presión diferencial sugirió realizar una separación de las tierras digitales y análogas. Ambas tierras se unen en un sólo punto mediante una resistencia de $0[\Omega]$.

Al ser estos sensores componentes rara vez utilizados y tener encapsulados poco estándares, fue necesario crear las librerías para EAGLE. También se creó la librería para el *opamp*.

Módulo IR

En internet es posible encontrar muchos diseños de sensores IR ya probados extensamente. El circuito utilizado corresponde al encontrado en la página web de “Paparazzi” [21]. Paparazzi es un proyecto libre de software y hardware para la implementación completa de un piloto automático. El diseño de los sensores IR ha sido mejorado a lo largo de los años y la versión utilizada en este proyecto es la más actual disponible a la fecha (versión de septiembre 2009).

El arreglo utilizado es el descrito en el capítulo anterior en la descripción de los sensores. Es necesario utilizar un *opamp* en configuración no inversora para amplificar la entrada y un filtro pasa bajos a la salida para eliminar las altas frecuencias que distorsionen la lectura. Fue necesario crear las librerías de EAGLE para los sensores y para el *opamp*.

Módulo PIC

El módulo PIC incluye:

- Microcontrolador.
- Condensadores de desacoplo.
- Oscilador.
- Voltajes de referencia para el módulo ADC.
- LED indicador de energizado del módulo.

Los microcontroladores de gama alta requieren de diversas fuentes de alimentación, en este caso 2:

- 2.5[V] para el núcleo.
- 3.3[V] para los periféricos en general.

Este microcontrolador posee un regulador interno para alimentar su núcleo. Si se desea se puede habilitar este regulador o utilizar uno externo. En este caso se decidió utilizar el regulador interno, por lo que es necesario incluir un condensador de poliéster como estabilizador del voltaje. La alimentación de los periféricos es provista por la línea de 3.3[V]. En las alimentaciones se colocan condensadores de desacople. Para aprovechar al máximo las capacidades del microcontrolador se utiliza un cristal de 32[MHz]. Gracias a este cristal el microcontrolador alcanza los 16 MIPS (millones de instrucciones por segundo) sin necesidad de activar el PLL.

Las referencias de voltaje para el módulo ADC son las mismas que la alimentación del microcontrolador (0 – 3.3[V]). Para aislar el módulo ADC de la línea de alimentación del microcontrolador, el fabricante, en el *datasheet* [29], recomienda unir la referencia positiva mediante una resistencia de 10[Ω].

Se ha incluido un LED, con una resistencia en serie, que indica si la placa se encuentra energizada o no, está simplemente conectado a la línea de 3.3[V].

Módulo GPS y RF

Los módulos GPS y RF son unidades que van montadas directamente en la placa madre y que para su correcto funcionamiento sólo es necesario añadir condensadores en sus alimentaciones. El consumo elevado y pulsante del XBee sugiere la utilización de un condensador de gran capacidad en la línea de alimentación (2200[uF]) y así estabilizar la alimentación y disminuir el ruido de esta línea para no afectar a los demás componentes del circuito. El condensador requerido por el GPS es cerámico y de sólo 1[uF] pues el consumo es muy bajo. Otros componentes añadidos son detallados en la descripción de la placa madre.

La recepción de los datos enviados mediante RF son recibidos en tierra por un kit de desarrollo, diseñado para los módulo XBee, que sirve para tanto para realizar la configuración de éstos como para conectarlos al PC mediante USB para recibir o transmitir información. Esta placa es parte del kit que se compró junto a los módulos XBee, por lo que no forma parte del desarrollo de la memoria.

Placa madre

La placa madre aloja todos los módulos anteriormente descritos menos el módulo comuta/decod. Para el diseño es necesario crear librerías de EAGLE de todos los módulos, pues éstos

son vistos desde la placa madre como cajas negras que realizan determinadas funciones y sólo es necesario conocer las conexiones con el exterior y no la estructura interna.

En la placa madre es posible encontrar los siguientes conectores de salida/entrada asociados a sus respectivos módulos:

- Alimentación principal (módulo alimentación).
- Puertos de expansión (ADC, PWM, INT, GPIO) (módulo PIC).
- Buses de datos I²C y SPI para aplicaciones futuras (módulo PIC).
- Conectores para sensores IR, tanto horizontal como el sin implementar vertical (módulo PIC).
- Conectores para alimentar circuitos externos con tensiones de 3.3[V] y 5.5[V] (módulo alimentación).
- Entrada de voltaje de alimentación del circuito receptor.
- Servos (módulo PIC).
- Puerto serial 1 y 2 (módulos PIC, GPS, RF).
- Conector para comunicación entre el microcontrolador y el módulo conmuta/decod.
- Conector para realizar la programación del microcontrolador.

Como se mencionó anteriormente, algunos elementos asociados al módulo RF se encuentran en la placa madre. El primero es el circuito de reset; compuesto por un transistor, cuya base está conectada a una salida del microcontrolador, en configuración colector abierto. El reset del XBee se produce cuando la entrada “reset” del módulo esta conectada a tierra por un período superior a 200[uS]. También se añadieron diodos protectores y resistencias *pull-down* en las líneas RX y TX. Las resistencias se utilizan para, en el caso de una desconexión entre el microcontrolador y el XBee, evitar que las líneas queden sin referencia y entreguen lecturas falsas.

Ambos puertos seriales son accesibles desde el exterior, pero sólo el puerto serial #1 (asociado al XBee) está diseñado con ese propósito en mente. En condiciones normales de funcionamiento es necesario realizar un puente en el conector del serial #1. Este puente cumple la función de unir las líneas TX y RX del XBee con las del microcontrolador. Sin embargo, para realizar un monitoreo

o debugeo del circuito en tierra, no es necesario estar conectado de manera inalámbrica. En estos casos se desconecta el puente y es posible acceder al puerto serial 1 del microcontrolador a través del mismo conector.

En la placa madre también se encuentra el circuito de reset del microcontrolador, las resistencias *pull-up* del bus I²C y los divisores de tensión para la lectura del voltaje de las baterías tanto de alimentación del receptor-servo como la alimentación de la placa madre. Estos divisores son necesarios, por encontrarse los voltajes fuera del rango de medición del microcontrolador.

Finalmente, se incluyeron 3 LEDs conectados a salidas del microcontrolador para debugeo y estado.

Convertor USB-Serial

En la hoja de datos del convertor serial-USB FTDI232, [30], se encuentran diversas configuraciones modelo para lograr la interfaz USB con dispositivos funcionando a 3.3[V], sólo falta añadir una inductancia en la entrada V+ del USB, 2 condensadores de desacoplo y un condensador para el regulador interno del integrado.

Con este convertor es posible acceder al puerto serial 1 del microcontrolador de manera alámbrica.

4.1.3. Encapsulados de los componentes utilizados en el prototipo

Dentro de los encapsulados disponibles para montaje superficial de resistencias y condensadores cerámicos se optó por utilizar los de tamaño intermedio 0805. Este encapsulado es de reducido tamaño, pero aún es posible de soldar a mano. Con un tamaño más pequeño, como el 0603, la resolución de la LPKF no permite colocar pistas entre los pads del componente de manera confiable.

Para los condensadores electrolíticos y de poliéster, se eligió el encapsulado más pequeño disponible, que fuera mayor a los 0805.

El resto de los componentes como LEDs, inductancias, diodos y transistores, tienen un tamaño similar a los componentes 0805.

Para los integrados, los encapsulados superficiales utilizados son:

- SOIC16 (Small outline integrated circuit), CD4053.
- SOP16 (Small outline package), CD4017.

- TSOP28 (Thin small outline package), FT232RL.
- S-PDSO-G8(Power pad plastic small outline package), TLV2472.

El único integrado no superficial es el PIC12F09 de la placa conmuta/decod. Se decidió utilizar un encapsulado DIP8, pues estando el componente soldado a la placa es necesario añadir un puerto de programación. Dado el reducido tamaño del PIC12F09 es más conveniente utilizar una base y desmontarlo cada vez que se desea programar.

4.1.4. Ruteos

La utilización de módulos reduce de manera considerable la complejidad que implica realizar el ruteo de tan elevado número de componentes, debido a que los componentes se agrupan en sus respectivos módulos en áreas delimitadas. La manera de trabajar es la misma para todos los módulos.

A partir del esquemático generado para cada módulo se obtiene una placa en la cual se encuentran todos los componentes representados por su forma física y todas las interconexiones de los componentes para cada nodo. Se decidió utilizar placas de 2 capas, por lo que es posible colocar componentes en la capa superior e inferior. La disposición original de los componentes no presenta ningún orden, por lo que el primer paso consiste en realizar una disposición en bruto y comenzar la transformación de los nodos en pistas. El diseño de un PCB es un proceso iterativo en el que hay que tener en cuenta una serie de restricciones y consideraciones. En primer lugar, es necesario conocer los límites de resolución de la LPKF:

1. La separación entre pistas no puede ser menor a 8[mil]. No se cuenta con herramientas que permitan realizar separaciones menores.
2. El ancho mínimo de las pistas es 10[mil].

El ancho de una pista depende de la función que cumple. La pista de alimentación principal no puede ser muy delgada pues por ella circularán elevadas corrientes y sus ramificaciones pueden ir disminuyendo su ancho. El largo de las pistas es también un factor a considerar, especialmente en el ruteo de señales analógicas. En determinados casos es necesario unir señales de una capa con la otra, esta operación se hace utilizando el menor número de vías posible.

Todas las placas cuentan con un plano de tierra, este plano no es más que el negativo de la alimentación y sirve para reducir interferencias y disminuir el número de pistas. El diseño se optimiza considerando los factores anteriores e intentando lograr una placa del menor tamaño posible, lo que se realiza reposicionando componentes y reorganizando pistas.

Finalmente se trazan los límites del PCB, que definen su tamaño real y servirán de guía para el corte final.

La siguiente es una imagen de la placa madre con los módulos (Figura 4.1.3). En azul se observan los componentes y pistas de la capa inferior (*Bottom*), en rojo los de la capa superior (*Top*) y en verde los pads.

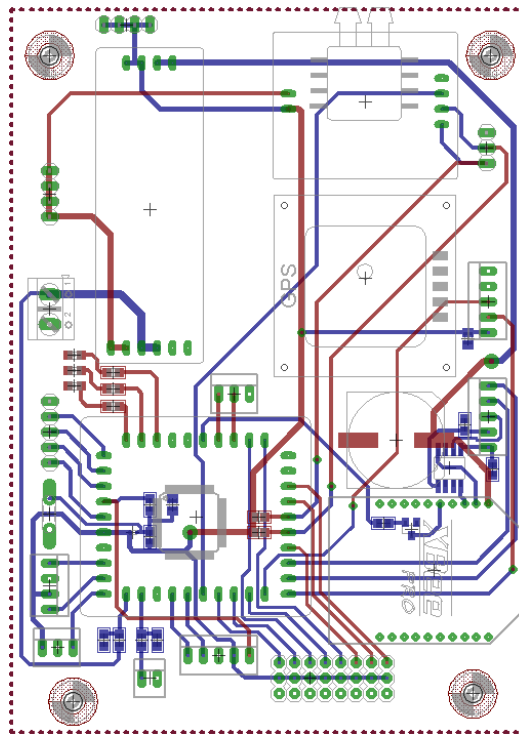


Figura 4.1.3: Ruteo de placa madre, escala 1:1.3[cm]

4.1.5. Construcción

A partir del ruteo, el mismo EAGLE® genera los archivos Gerber necesarios para la construcción del PCB. Las capas importadas en el CircuitCAM® son:

- Capa superior.
- Capa inferior.
- Aperturas.
- Borde.
- Listado de herramientas.

Para una adecuada confección de las capas superior e inferior es necesario seleccionar que herramientas se van utilizar y definir el aislamiento de las pistas. Con las herramientas y aislamiento se procesan las trayectorias a seguir para remover el cobre necesario y de esta forma grabar las pistas. Dado el tamaño de algunos componentes es necesario utilizar varias herramientas:

- Micro Cutter 0.1[mm].
- Universal Cutter 0.2[mm].
- End Mill RF 0.4[mm].

El programa siempre sugiere utilizar la herramienta de mayor tamaño en todo lugar donde sea posible. La Micro Cutter 0.1[mm] se utiliza para los detalles más pequeños, la Universal Cutter 0.2[mm] para el contorno de algunas pistas y la End Mill RF 0.4[mm] para la remoción del resto de cobre.

La siguiente capa a procesar es la de los taladros. Las herramientas utilizadas para realizar los taladros son seleccionadas automáticamente a partir de una lista que especifica las herramientas disponibles.

Finalmente, se procesa la capa que incluye el corte externo de la placa. Para esta operación se seleccionó la herramienta de corte Countour Router de 2[mm] de diámetro.

Con todas las etapas procesadas, se realiza una inspección visual de los recorridos en las áreas más sensibles para comprobar que hay una herramienta que realiza el trabajo esperado de aislación,

como por ejemplo en las separaciones de 8[mil]. El proyecto es finalmente exportado como un archivo al BoardMaster®.

El orden que se siguió en el proceso de construcción del PCB en la LPKF fue: capa superior, capa inferior, taladro y corte externo. La LPKF es una fresadora que tiene actuadores en los ejes X Y Z, pero sólo X e Y se mueven en todo el espacio de la mesa. El eje Z tiene únicamente dos posiciones: abajo o arriba. El mecanizado de las distintas placas utiliza las herramientas definidas en el CircuitCAM®, las que deben ser reemplazadas a mano. En este punto se aprecia la más importante debilidad de la LPKF: la necesidad de ajustar la profundidad de corte de cada herramienta manualmente, bajando o levantando el husillo, utilizando como criterio la experiencia. Si la herramienta se coloca muy abajo, el corte será muy profundo. Al utilizar herramientas con filo cónico (Micro y Universal Cutter), a mayor profundidad de corte, mayor ancho del desgaste del cobre. En cambio, si la herramienta se coloca muy arriba, pueden quedar sectores donde no se alcance a traspasar el cobre hasta el dieléctrico con la consecuente falla en el aislamiento. Es preferible trabajar con la herramienta más arriba, pues en caso de ser necesario se pueden reparar los sectores que presenten problemas. Con las herramientas cilíndricas, como las End Mill RF, no se tiene esta subjetividad respecto al ancho de las pistas. Sin embargo, el desgaste de la herramienta implica realizar constantes correcciones en su altura para conseguir resultados satisfactorios.

Al finalizar todo el proceso de mecanizado se obtienen PCBs como las de la Figura 4.1.4:

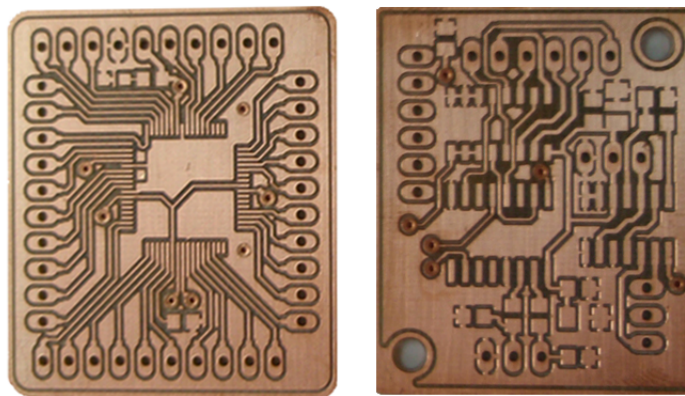


Figura 4.1.4: Ejemplos de PCBs mecanizados y listos para el soldado de componentes

4.1.6. Revisión de PCBs y soldado de los componentes

Antes de realizar el soldado de los componentes en las placas, es necesario revisarlas. Al igual que cualquier otro proceso de mecanizado, se deben eliminar las rebabas de material. Las rebabas se remueven utilizando virutilla Scotch Brite® y un bisturí. La virutilla también elimina los aceites y óxidos formados en la superficie del cobre debido a la manipulación. A continuación, se insertan y remachan todas las vías.

La última revisión antes del soldado consiste en comprobar continuidad entre pistas. Primero se verifica que no exista conexión alguna entre el plano de tierra con el positivo de la alimentación o con alguna otra pista. Después, que no exista conexión entre pistas aledañas. También se comprueba continuidad en extremos de pistas muy largas.

Los componentes deben ser soldados en la placa y dado los tamaños es necesario usar estaciones de soldado especiales para componentes superficiales. Se utilizó una estación ERSATM disponible en el laboratorio de electrónica del departamento. Esta estación permite regular con exactitud parámetros como temperatura en la punta del cautín e inercia térmica de la punta. Para la soldadura de componentes SMD se utilizó la punta “mini ola” y para los *through hole* puntas de alfiler y paleta. La temperatura de soldado, tanto para SMDs y *through hole*, fue de 320[°C], que está dentro del rango de temperatura de todos los componentes y es suficiente para trabajar con soldadura fluida. Aquí también se encuentran los demás materiales utilizados en el proceso de soldadura: estaño, pinzas, alicates, cortantes y flux. El flux es una pasta que ayuda a que el estaño fluya y se adhiera sólo a los pads de cobre y no cortocircuite líneas. De esta forma, es posible soldar sin problemas integrados como el microcontrolador.

Como regla general en el soldado de componentes, el orden depende del tamaño del componente:

1. Componentes de menor tamaño SMD: resistencias, condensadores cerámicos, inductancias, transistores, LEDs, circuitos integrados condensadores poliéster y electrolíticos.
2. Componentes de mayor altura: conectores.

De esta forma, se evita que la mayor altura de los conectores interfiera con el soldado de los componentes de menor tamaño.

4.1.7. Pruebas de funcionamiento de los módulos

En la etapa de pruebas se busca comprobar el correcto funcionamiento de todos los elementos que componen la placa. La estructura modular facilita esta tarea ya que es posible testear los módulos por separado y así evitar daños en el resto de la placa en caso de un mal funcionamiento.

El primer módulo puesto a prueba es el módulo de alimentación. Con sólo el módulo de alimentación montado en la placa madre se procede a energizar el prototipo utilizando una fuente de voltaje y corriente regulados. La opción de limitar corriente desde la fuente resulta muy útil, pues en caso de existir un cortocircuito se pueden impedir daños mayores a los componentes y pistas. En funcionamiento, la alimentación será una LiPo cuyo voltaje puede variar entre 7.4 - 8.4[V], el módulo es alimentado con todo el rango de voltajes posible. Una vez descartado un cortocircuito se miden los voltajes de las distintas líneas de alimentación comprobando que sean las esperadas (3.3, 5.0 y 5.5[V]).

El siguiente módulo a probar es el del microcontrolador. En este momento son de utilidad los LEDs presentes en la placa madre, pues una forma de probar el funcionamiento del microcontrolador es realizando un simple programa de prueba que apague y encienda los LEDs. Otro test realizado consiste en establecer comunicación a través del puerto serial #1 con el computador. Con esta prueba también es posible verificar que los distintos parámetros de configuración inicial (fusibles) del microcontrolador han sido establecidos correctamente, en especial los relacionados al reloj. Con el puerto serial funcionando correctamente se procede a realizar lecturas del ADC conectado a la alimentación de la placa madre. Imprimiendo en pantalla el valor obtenido de la conversión y variando el voltaje de alimentación de la placa madre es posible comprobar el correcto funcionamiento de los puertos ADC. El puerto serial #1 es el medio utilizado para testear el resto de los módulos.

Como se detalló anteriormente, el GPS se comunica con el exterior enviando sentencias NMEA. Utilizando el conversor serial-USB es posible comunicarse directamente con el módulo GPS. En una primera instancia es necesario configurar al GPS con la velocidad del puerto serial a utilizar, las sentencias a enviar y la frecuencia del envío de éstas. Utilizando las sentencias de configuración entregadas por ETEK Navigation en [31] y un programa monitor del puerto serial se configura el GPS para transmitir por el puerto serial a 56760[bps], enviar solamente la sentencia RMC y realizar el envío con una frecuencia de 1[Hz]. La sentencia RMC (Recommended Minimum Specific

GPS/TRANSIT Data) es la que entrega toda la información necesaria para realizar la navegación. Un ejemplo de la sentencia se muestra a continuación:

\$GPRMC,123519,A,4807.038,N,01131.000,E,022.4,084.4,230394,003.1,W*6A

Donde:

Tabla 4.1.1: Ejemplo de los campos de una sentencia NMEA

\$	Inicio de la sentencia
RMC	Recommended Minimum sentence
123519	Fix hecho a las 12:35:19 UTC
A	Estado A=activo o V=no activo
4807.038,N	Latitud 48 grados 07.038' N
01131.000,E	Longitud 11 grados 31.000' E
022.4	Velocidad sobre tierra en nudos
084.4	Rumbo verdadero [°]
230394	Fecha – 23 de marzo 1994
003.1,W	Variación magnética
*6A	Checksum data, siempre comienza con * (XOR de todos los caracteres)

Con el GPS configurado, se monta en la placa madre y se conecta con el puerto serial #2 del microcontrolador. El microcontrolador se comporta simplemente como un puente entre el puerto serial #2 y #1. Lo que recibe del GPS por el puerto serial #2 lo envía por el puerto serial #1. Alimentando la placa madre con la LiPo y conectado a un notebook es posible desplazarse y observar como cambian los campos de la sentencia RMC, lo que ratifica el correcto funcionamiento del módulo GPS.

El microcontrolador recibirá entonces esta sentencia RMC compuesta sólo por caracteres. Para poder trabajar con los datos es necesario hacer una interpretación de la sentencia y una conversión de texto a números. El resultado se almacena en variables globales que pueden ser leídos por cualquier función. Se realiza la misma prueba anterior, pero esta vez imprimiendo en pantalla sólo los campos: latitud, longitud, rumbo y velocidad. De esta forma se comprueba el correcto funcionamiento de la función que realiza la interpretación de la sentencia recibida.

El funcionamiento de los sensores IR se comprobó en una primera instancia, energizándolos y midiendo con un voltímetro la salida. Se apuntaron los sensores al cielo y a la tierra y se observó si variaba el voltaje de salida. A continuación se conectaron las salidas al microcontrolador. Los valores obtenidos de los canales ADCs a los que se encuentran conectadas las salidas del sensor son enviados a través del puerto serial #1 al computador. Utilizando MATLAB® se escribió un

sencillo código que almacena los datos recibidos por el puerto serial durante un período de tiempo determinado. Finalmente, los datos son graficados. Un ejemplo del resultado obtenido al mover la mano frente a un par de sensores se muestra a continuación en la Figura 4.1.5:

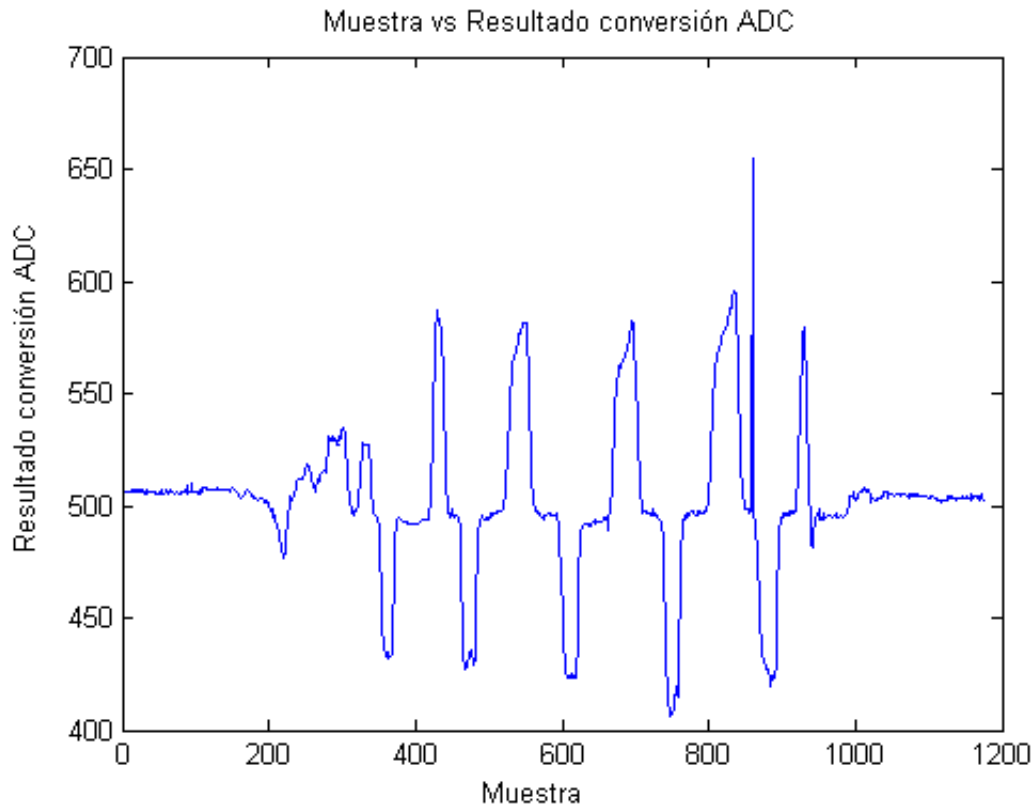


Figura 4.1.5: Prueba de los sensores IR

El módulo presión fue el que presentó mayores complicaciones. El sensor Bosch™ es un dispositivo que se comunica utilizando el protocolo I²C y su funcionamiento se basa en la escritura y lectura de registros en un mapa de memoria. Los osciloscopios disponibles en el laboratorio de electrónica tienen una función de lectura de bus I²C que consiste básicamente en el monitoreo de las señal SCL y SDA. Se escribió un sencillo programa que se intentará comunicar con el sensor, simplemente se escribe la dirección del sensor en el bus y se espera la respuesta del sensor. Al no recibir respuesta alguna se procedió a re-soldar el componente (encapsulado LCC8), lo que dio buen resultado pues se logró establecer la comunicación con el sensor. A continuación, se realizaron lecturas de diferentes registros y se imprimieron en pantalla a través del puerto serial #1. Las lecturas varían al ser sometidas a cambios de presión, lo que sugiere un correcto funcionamiento.

El sensor de presión diferencial tiene salida analógica, por lo que basta leer el valor obtenido en el ADC al cual se encuentra conectado aplicando presión en las entradas del sensor y observando la salida. La utilización de ambos sensores de presión requiere de una calibración antes de poder utilizar los datos que entregan. Para el caso del sensor Bosch™, la calibración consiste en la lectura de una serie de constantes que se utilizan en la función que calcula la presión. Para el caso del sensor de presión diferencial la calibración consiste en realizar un experimento con un instrumento calibrado, comparando lecturas a distintas velocidades y generando una LUT.

Finalmente se prueba el módulo conmuta/decod. Conectando el módulo entre un receptor y un servo se comprueba el correcto funcionamiento. Se escribe un programa de prueba para el microcontrolador que genera un pulso que crece y decrece en el tiempo y una señal de reset. Para el PIC12F509 utilizado en el módulo se escribe un programa que mide el ancho del pulso enviado por el receptor y en base a este ancho dispara o no la conmutación. El diagrama de flujo (Figura 4.1.6) de este programa es el siguiente:

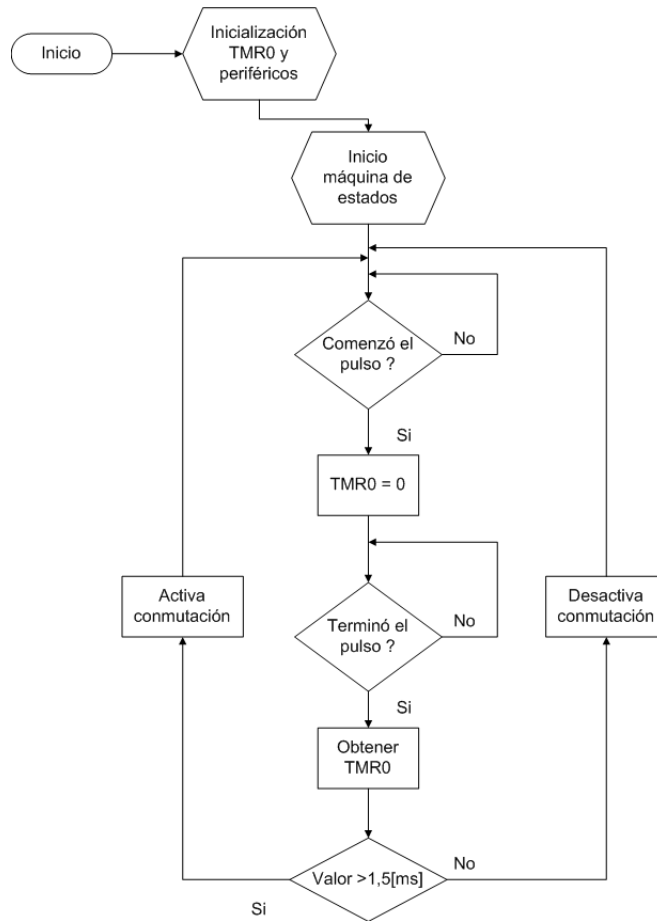


Figura 4.1.6: Diagrama de flujo del programa del módulo conmuta/decod

La interferencia a la que a veces se ve sometido el receptor puede activar o desactivar la conmutación sin que la orden sea real. Por este motivo, antes de activar o desactivar la conmutación, se realiza un conteo de pulsos del estado deseado que debe ser mayor a un cierto umbral para que se ejecute la conmutación. La conmutación tarda entonces algunos [ms] en ser detectada desde el momento de la activación en el radiocontrol. Este umbral de pulsos válidos se puede disminuir o incrementar si se desea.

La prueba del módulo conmuta/decod arrojó los resultados esperados: en modo RC el servo se mueve según lo comanda el radiocontrol y al activar la conmutación el servo se mueve según lo ordena el microcontrolador. El umbral del conteo de pulsos se fijó en 25.

La placa madre con todos sus módulos montados y el sensor IR se muestran en la siguiente imagen (Figura 4.1.7):

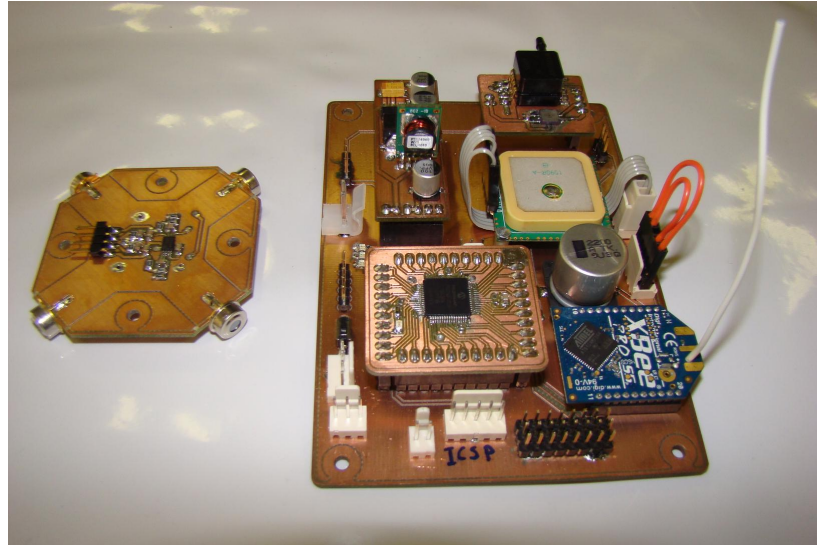


Figura 4.1.7: Electrónica soldada

4.2. Construcción del aeroplano

En esta sección se muestran las etapas que comprendieron la construcción y pruebas del aeroplano. También se detallan los materiales y técnicas de construcción utilizadas. La construcción de un aeroplano de estas características y dimensiones requiere de un lugar en el cual no exista intervención de terceros, los tiempos de secado de pegamentos y fragilidad de las piezas que no están terminadas son muy sensibles a manipulaciones indebidas. El memorista cuenta con un taller en el que se encuentran todas las herramientas necesarias.

4.2.1. Programa utilizado

SolidWorks® 2007 (desarrollado por SolidWorks Corporation™)

Es un programa de diseño asistido por computador (CAD) para modelado mecánico. En el se pueden modelar piezas y conjuntos y extraer de ellos tanto planos como otro tipo de información necesaria para la producción. El proceso consiste básicamente en ir creando piezas que después son ensambladas unas con otras hasta formar el modelo.

4.2.2. Modelo computacional

En el Capítulo 3.2 se detalló el cálculo de las dimensiones y masa del aeroplano. A partir de estos cálculos se construye un modelo computacional del aeroplano. El modelo brinda la posibilidad de observar detalles constructivos y de dimensiones que no son apreciables en los cálculos. A partir de estas observaciones se realizan algunas modificaciones en los cálculos hasta que se logre el resultado esperado, es un proceso iterativo en el que se busca un adecuado compromiso entre los cálculos preliminares y el modelo.

El primer componente del modelo fue el ala, se importó el perfil a utilizar y se creó el ala con las dimensiones especificadas en los cálculos, luego se dibujaron los planos de cola: estabilizadores horizontal y verticales, finalmente se diseñó un fuselaje a partir de las medidas obtenidas en los cálculos. Sólo se calcularon dimensiones brutas del fuselaje, la forma se definió tomando en consideración el tamaño del motor, estanque de combustible, electrónica de automatización y un buen anclaje para el ala.

El resultado se muestra en la Figura 4.2.1:

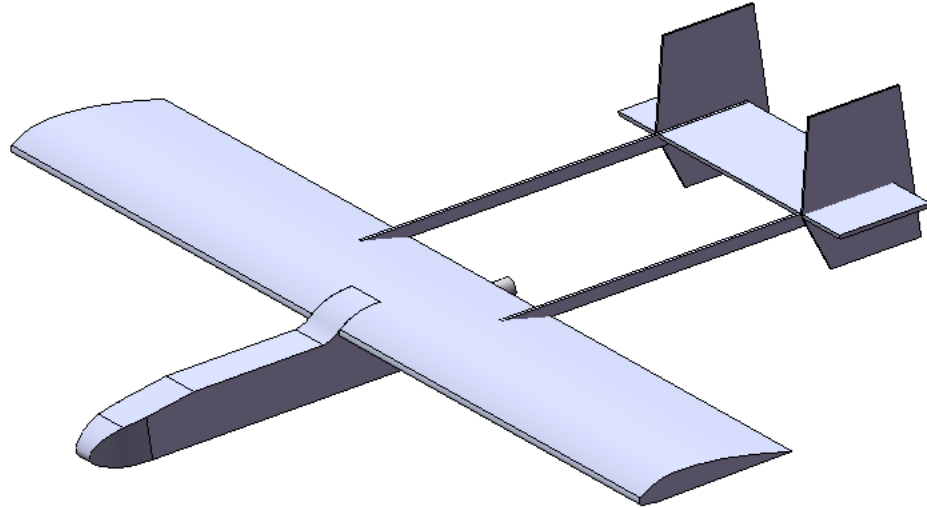


Figura 4.2.1: Modelo computacional del aeroplano

4.2.3. Materiales, técnicas y construcción

Fuselaje

Hay muchas técnicas de construcción para un fuselaje, la más común es una estructura tipo cajón de madera reforzado con cuadernas. Este tipo de construcción limita el espacio interior ya que crea compartimientos a los que puede ser difícil acceder en un fuselaje de grandes dimensiones. Para superar este problema se decidió construir el fuselaje enteramente en fibra de vidrio y utilizando cuadernas sólo en los lugares donde fuera necesario, como fijación del estanque de combustible y tren de aterrizaje delantero.

La fibra de vidrio es parte de la familia de los materiales compuestos. Los materiales compuestos son aquellos que se forman por la unión de dos materiales para conseguir propiedades que no es posible obtener en los materiales originales. Estos compuestos pueden seleccionarse para lograr combinaciones poco usuales de rigidez, resistencia, peso, rendimiento a alta temperatura, resistencia a la corrosión, dureza o conductividad.

Los materiales compuestos reforzados con fibras están divididos en dos componentes: el primero es un agente de refuerzo como una fibra fuerte: fibra de vidrio, kevlar o fibra de carbono que proporciona al material su fuerza a tracción, y el segundo (llamado matriz) suele ser una resina

como epoxy o poliéster que envuelve y liga las fibras, transfiriendo la carga de las fibras.

Para la fabricación de una pieza en fibra de vidrio se puede usar un molde o simplemente cubrir de fibra un modelo. El método basado en moldes es más utilizado si se quiere hacer la pieza varias veces, pero requiere de la fabricación de un modelo con terminaciones perfectas (el “positivo”) que luego se cubrirá de fibra. Una vez curada la resina se divide la cobertura en varias partes sacando el positivo, dejando un “negativo”. El negativo es cubierto luego con fibra y resina, se unen las distintas piezas y finalmente se obtiene un modelo que es exactamente igual al positivo original. Este método tiene muy buenos resultados, pero la generación de un molde eleva mucho los costos.

Para el presente caso, la construcción de un sólo prototipo, no es necesario aplicar el método del positivo-negativo. La técnica que se usó consiste básicamente en la fabricación de un modelo que es cubierto de fibra y del cual finalmente se extrae el molde interior, dejando sólo la “caparazón” de fibra.

En ambas técnicas, el proceso de colocar una capa de fibra de vidrio y luego impregnarla en resina se llama “laminado”.

El modelo se fabricó en plumavit (poliestireno expandido) de baja densidad ($10[\frac{kg}{m^3}]$). La ventaja de utilizar plumavit radica en la facilidad con que se puede trabajar y que al finalizar el laminado se puede deshacer el molde utilizando diluyente o simplemente extrayendo pedazos a través de un orificio practicado al fuselaje.

Para lograr un modelo lo más fiel posible al diseñado en SolidWorks® fue necesario imprimir el modelo. El perfil de papel es entonces copiado en dos láminas de aluminio. La sencilla geometría del fuselaje permite que con una sólo postura sea posible obtener la forma deseada. Se corta un bloque de plumavit de dimensiones mayores (alto y largo) al paralelepípedo que encierra al modelo, se fijan los perfiles de aluminio en ambos lados del bloque y utilizando un arco, cuyo tensor es un cable de nicrom, se corta el perfil en la plumavit. El cable de nicrom es utilizado como calefactor en secadores de pelo y anafres, al ser tensado y conectado a una fuente de voltaje, se calienta y permite realizar cortes con gran precisión en la plumavit.

Utilizando una plantilla de papel y un corta cartón se le da la forma deseada a la punta del fuselaje. El corte con nicrom deja una superficie relativamente lisa que es pulida con lija de grano fino (#150).

El tipo de resina utilizado en la fabricación del fuselaje es epóxica. La resina de poliéster es

más fácil de trabajar, pero es más tóxica, deshace la plumavit y no tiene propiedades mecánicas tan buenas como la epóxica. Si se realiza el laminado directamente sobre la plumavit, la resina se adherirá, lo que dificultará el proceso de extracción de la plumavit y dejará el interior del fuselaje con una textura rugosa. Para evitar este problema se cubrió todo el modelo con capas de resina epóxica que una vez curada fue lijada. El modelo es cubierto con una capa de cera desmoldante que facilita la separación entre el molde y el fuselaje.

La resina epóxica utilizada es especial para el proceso de laminado, pues es de muy baja densidad, lo que permite una fácil aplicación y rápida absorción. La fibra de vidrio utilizada es el tejido tipo “Woven Roving” de $160[\frac{g}{m^2}]$.

Antes de comenzar el laminado es necesario dimensionar los paños de tejido dependiendo de la posición que ocuparán en el modelo. La resina epóxica comienza a endurecerse a los 30 minutos de que ha sido mezclada con el endurecedor, por lo que el tiempo de trabajo es reducido. El proceso seguido es el siguiente:

1. Se cortan los paños.
2. Se realiza la mezcla resina-endurecedor.
3. Se posiciona el paño sobre el modelo y se aplica resina con una brocha, procurando impregnar adecuadamente la fibra.
4. Se envuelve el modelo con delgadas películas de plástico para copiar de la mejor manera posible los detalles.
5. Se espera que la resina cure completamente (aprox 30 horas).
6. Se retiran las láminas de plástico y se lija todo el modelo para permitir la correcta unión con la siguiente capa.
7. Se repite el proceso.

Se prestó especial atención en los lugares que sufren los mayores esfuerzos:

- Tren de aterrizaje trasero y delantero.
- Anclaje de motor.
- Anclaje de ala.

En estos puntos de esfuerzo se puso un mayor número de capas traslapadas de manera tal que el grosor del laminado creciera gradualmente y así se pudiera distribuir la energía uniformemente. El mínimo número de capas en todo el fuselaje es 6, y en las partes de mayor esfuerzo, como en la fijación del motor, el número aumenta hasta 12.

Uno de los objetivos del diseño pusher era contar con sectores separados; electrónica de automatización y radio, motor y estanque de combustible. Se realizaron dos cortes en el fuselaje que permiten un adecuado acceso a todo el fuselaje y a los citados sectores. El primero se encuentra bajo el punto de sujeción del ala y sirve para acceder al estanque de combustible, motor y electrónica del motor. El segundo se encuentra en la parte anterior del fuselaje y brinda acceso desde la punta del fuselaje hasta el inicio del estanque de combustible. El diseño del fuselaje con los elementos en su interior se muestra a continuación en la Figura 4.2.2:

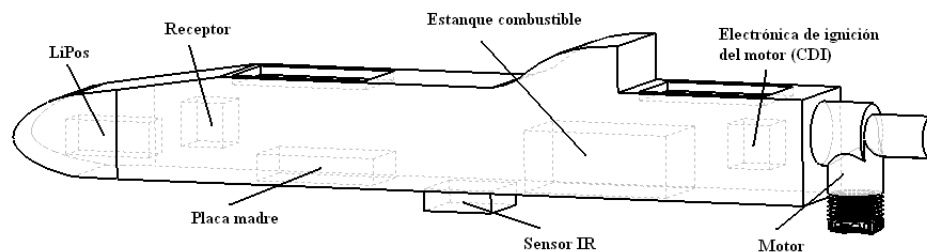


Figura 4.2.2: Distribución de los elementos en el fuselaje

A través de los cortes hechos en el fuselaje fue posible retirar la plumavit del interior sin tener que utilizar diluyente. La Figura 4.2.3 muestra una porción del fuselaje antes y después de ser enfibrado:

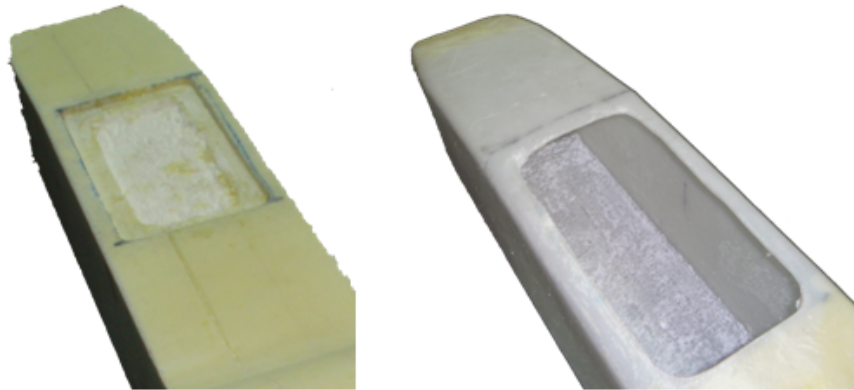


Figura 4.2.3: Fuselaje pre y post fibra

Con el fuselaje libre de plumavit se reforzaron con madera terciada de 3[mm] los lugares sujetos a mayor estrés. Para lograr un buen acabado, se lija el exterior con lija fina (#200) y se aplica masilla en los lugares donde sea necesario emparejar.

Alas

El diseño pusher implica que el centro del ala es el único punto en que el fuselaje tiene contacto con todos los elementos sustentadores, el ala sirve de anclaje para los planos de cola. Por este motivo, y dada la gran envergadura del aeroplano es necesario diseñar un ala muy resistente. Los materiales utilizados en la construcción del ala son: plumavit de alta densidad, madera de balsa de 1 y 6[mm], madera terciada de 2 y 3[mm], tubos de fibra de carbono de 20[mm] de diámetro exterior, tubos de carbono de 6[mm] de diámetro exterior, varillas de carbono de 5,6[mm] diámetro exterior, perfiles de carbono de 5[mm] de lado, tubo de fibra de vidrio de 20[mm] diámetro interno y tela de fibra de vidrio de $160[\frac{g}{m^2}]$.

Un ala de 3[m] de envergadura dificulta mucho el transporte y como los planos de cola van anclados mediante los booms (booms: ver pág 80) al ala se decidió dividir el ala en 3 paneles:

- Panel central, envergadura equivalente a la separación de los booms.
- 2 paneles externos, un panel a cada lado del panel central.

El núcleo del ala esta hecho de plumavit de alta densidad ($20[\frac{kg}{m^3}]$). La técnica utilizada para realizar el corte en la plumavit es la misma descrita en la construcción del fuselaje. Sin embargo,

debido a la envergadura y escaso espesor de cada panel hay que prestar especial cuidado en realizar los cortes de los bloques sobre una superficie plana para evitar reviradas en el ala. El corte se hace teniendo en cuenta que el ala será enchapa en madera de balsa de 1[mm] de grosor. Sin esta consideración, el perfil alar resultante sería distinto al deseado. La Figura 4.2.4 muestra el ala cortada a partir de un bloque de plumavit.

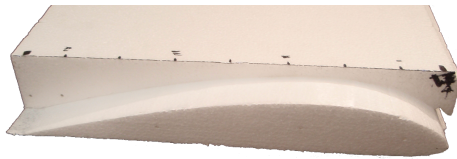


Figura 4.2.4: Corte transversal del ala

Los paneles se unen entre sí utilizando un sistema de bayonetas, es decir que en cada panel hay un conjunto de tubos-varillas que encajan en el conjunto de varillas-tubos del otro panel. La resistencia de la plumavit no es suficiente para soportar las fuerzas ejercidas por las bayonetas, por este motivo se insertaron costillas de madera de balsa y balsa-terciado en distintos puntos del ala. En la Figura 4.2.5 se muestra el diseño del ala con las varillas de refuerzos tanto en el intradós como en el extradós y las costillas (panel central y panel derecho):

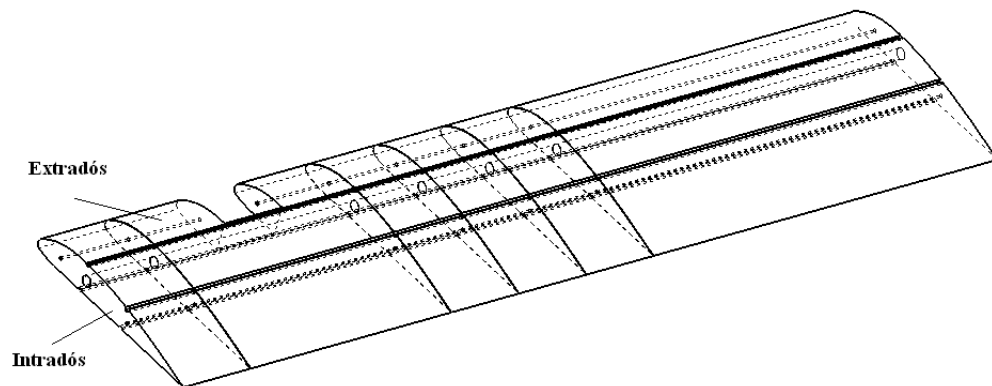


Figura 4.2.5: Plano estructural del ala

Las costillas del interior del ala son de madera de balsa y las que están en los puntos de unión de paneles son de balsa con terciado. La balsa se marca y pica fácilmente, no así el terciado. En la Figura 4.2.6 de la izquierda se aprecian las costillas y bayonetas de la unión del panel central con el panel exterior derecho.

En el panel central se realizaron refuerzos en terciado de 3[mm] en los sectores de anclaje del ala con el fuselaje a fin de distribuir la presión ejercida por los tornillos y bayonetas de anclaje sobre una mayor superficie. Con los refuerzos, costillas y bayonetas pegados y comprobado la correcta alineación se procede al trimado del borde de fuga y lijado de todo el conjunto para obtener una superficie lo más regular posible. El enchapado se realizó con balsa de 1[mm]. A los tableros de balsa es necesario aplicarles un sellador de poros en el lado que se pegará al ala. Sin este sellador la balsa absorbe grandes cantidades de pegamento, lo que impide una correcta unión con la plumavit. El pegamento utilizado para realizar la unión es la misma resina epóxica utilizada en el proceso de laminado del fuselaje. Se aplica una delgada película de resina sobre las chapas de balsa de intradós y extradós utilizando una espátula. Los restantes de plumavit del corte del ala son un negativo perfecto del perfil alar, en estos restantes se colocan las alas enchapadas y sobre ellos una masa cercana a los 50[kg] para hacer presión y lograr que las chapas queden unidas en toda la superficie. El panel central se reforzó con fibra de vidrio de 160[$\frac{g}{m^2}$] antes de ser enchapado. La Figura 4.2.6 muestra en el lado izquierdo el sistema de bayonetas y en lado derecho el panel central con sus refuerzos (sin fibra) y la chapa de balsa:

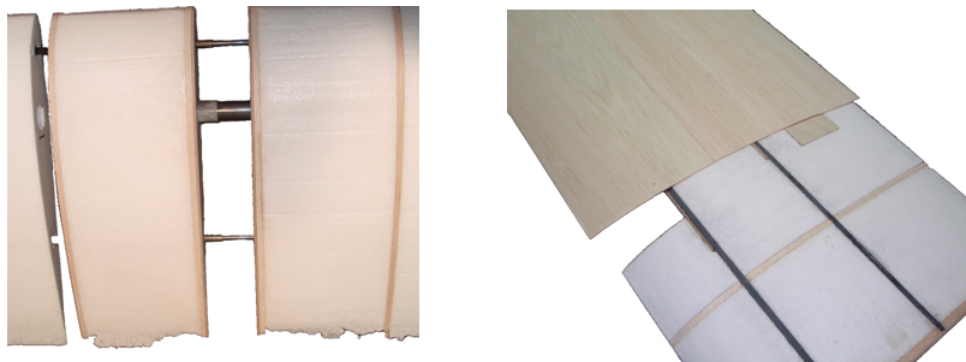


Figura 4.2.6: Sistema de bayonetas y chapa de balsa

El siguiente paso en la construcción de las alas es el refuerzo del borde de ataque y el corte de los alerones. El refuerzo del borde de ataque se hace cortando 1.5[cm] del actual borde de ataque y pegando en su lugar un perfil de balsa dura, el cual es necesario lijar para que obtenga la forma del perfil. El corte de los alerones se hace según las dimensiones calculadas de superficies de control. Los alerones se unirán al ala utilizando bisagras. La plumavit que queda expuesta al realizar el corte de los alerones no brinda el soporte adecuado para la fijación de las bisagras. Este sector del

ala y toda la envergadura de los alerones se refuerza con un listón de madera de balsa. El proceso es el mismo realizado en el borde de ataque.

Planos de cola

Los planos de cola (estabilizador horizontal y estabilizadores verticales) son construidos enteramente en balsa dura de 10[mm]. Se cortan según el plano diseñado y se perfilan para disminuir su resistencia al aire. Los planos de cola forman un conjunto no separable para el transporte, esto para aumentar su resistencia. Se unen a los booms mediante tornillos plásticos que pasan a través del estabilizador horizontal. El sector por el cual pasan los tornillos es reforzado con terciado de 2.5[mm] para evitar que la madera de balsa se venza y adquiera juego. Los estabilizadores verticales se encajan en el estabilizador y son pegados con resina epóxica de secado rápido. Un listón de madera de balsa refuerza la unión. Al igual que los alerones, las superficies de mando de los planos de cola se unen a las superficies fijas utilizando bisagras.

Booms

Los booms unen las alas con los planos de cola. A través de ellos se transfiere toda la fuerza que ejerce la cola a las alas. Para la construcción de los booms se utilizó madera de balsa dura de 15[mm]. Se aumentó el espesor en el área donde se une el boom con la cola y al igual que en el estabilizador horizontal, se reforzaron los puntos de contacto con madera terciada de 3[mm]. Los booms cubren toda la cuerda del ala y se fijan a ella simplemente por la presión que mantiene unidos los paneles de las alas. Se utilizan las bayonetas del ala para bloquear los booms en su lugar. Es necesario prestar especial atención a la posición en que quedan bloqueados: el ángulo de incidencia de la cola con respecto al ala debe ser cero. La Figura 4.2.7 muestra el detalle del bloqueo de los booms con las alas y con la cola:



Figura 4.2.7: Detalle boom y cola

Equipo de radio

Un diagrama eléctrico del aeroplano se muestra a continuación en la Figura 4.2.8:

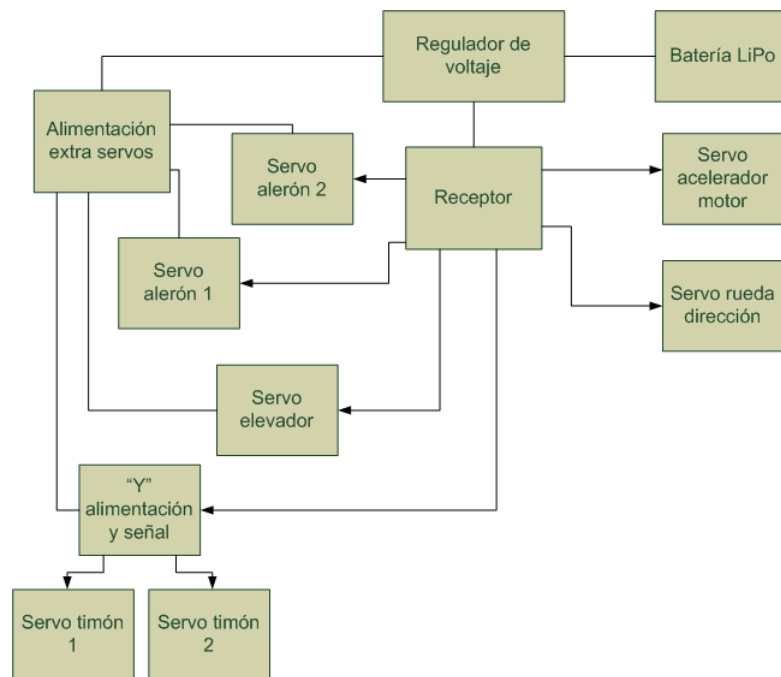


Figura 4.2.8: Diagrama eléctrico equipo de radio

Las superficies de mando tienen tamaños considerables y es necesario dimensionar correctamente los servos a utilizar. No hay problema en utilizar servos sobredimensionados, pero si el subdimensionado. El bajo torque puede destruir los servos o producir comportamientos no deseados,

lo que en vuelo es desastroso. Las recomendaciones para aviones de dimensiones similares son servos que tengan un torque mínimo de $100[\frac{oz}{in}]$. A mayor torque, mayor masa y tamaño del servo. En el control de superficies de mando son necesarios servos de alto torque; alerón #1 y #2, timón #1 y #2 y elevador. Para estas funciones se eligieron servos digitales, de la marca Hitec modelo HS5625. El torque de estos servos es de $110[\frac{oz}{in}]$. Los servos análogos son más económicos que los digitales, pero estos últimos tienen un torque más elevado que los análogos del mismo tamaño. El elevado torque que pueden realizar estos servos requiere de una alimentación independiente a la que puede proveer el receptor, entonces, sólo la señal de control proviene del receptor. Los dos servos de los timones se mueven al unísono. Con el fin de disminuir el número de cables que van hacia la cola se decidió instalar una “Y” para estos dos servos. Una “Y” recibe sólo una señal y alimentación en su entrada, las duplica y se obtienen en su salida dos señales y dos alimentaciones. La “Y” tiene que se dimensionada para el consumo de los servos.

Hay dos servos que no tienen requerimientos especiales de torque o dimensiones: el servo de la rueda de dirección y el servo del acelerador del motor. Se eligió un servo estándar análogo para estas tareas: Hitec HS422 ($45[\frac{oz}{in}]$). Estos servos reciben la señal y la alimentación directamente del receptor. La mayor diferencia entre servos análogos y digitales es la frecuencia con la que ejecutan el lazo de control de posición. En los servos análogos esta frecuencia es de $50[\text{Hz}]$, en los servos digitales puede alcanzar hasta los $300[\text{Hz}]$.

El receptor elegido es también marca Hitec modelo Supreme IIS de 8 canales.

La alimentación proviene de una LiPo de $7.4[\text{V}]$ y con una capacidad de $2800[\text{mA}]$. Esta LiPo está conectada a un regulador lineal capaz de entregar hasta $8[\text{A}]$.

Terminaciones

La etapa de las terminaciones incluye el anclaje del motor y estanque de combustible al fuselaje, el anclaje del tren delantero y trasero, el anclaje del ala al fuselaje, la instalación del equipo de radio, el pintado del fuselaje y el *monokoteado* de las alas y cola.

El motor se fija al cortafuegos del fuselaje utilizando pernos y tuercas. El diseño contempla al motor montado con ángulos de incidencia (antipar y picado) iguales a 0° con respecto a las alas. La posición del eje de empuje del motor está muy cercana al eje del ala, lo que posibilita esta configuración de ángulo nulos. Para su funcionamiento, el motor utiliza un CDI (capacitive discharge

ignition) para generar la chispa en la bujía. El CDI necesita una alimentación regulada entre 5.0 y 6.0[V]. La experiencia indica que no es recomendable utilizar la misma fuente de alimentación para el CDI y la radio, por lo que se incorpora una nueva LiPo de 7.4[V] para alimentar este consumo. Como el voltaje de alimentación debe ser regulado, se instala un regulador lineal entre la LiPo y el CDI. Los CDI también son conocidos por generar interferencia en las líneas de señales de los servos, por este motivo se montó el conjunto CDI-regulador en la parte posterior del fuselaje muy cercano al motor y alejado del equipo de radio. El estanque de combustible es fijado al fuselaje mediante 3 cuadernas de madera de balsa de 6[mm]. Las posiciones del estanque, motor y CDI se observan en la Figura 4.2.2. La LiPo que alimenta al CDI va montada junto a la LiPo del equipo de radio en la punta del fuselaje.

El tren de aterrizaje trasero está hecho en aluminio de 3[mm] de espesor. La posición en la que va montado corresponde a la indicada por los cálculos y se fija utilizando pernos y tuercas. El tren de aterrizaje delantero está hecho de acero de resortes en 4[mm] de diámetro. El tren delantero es móvil, para poder manejarlo y fijarlo es necesario pegar una cuaderna dentro del fuselaje que tenga los anclajes para la varilla de acero y para el servo de dirección.

El ala se ancla al fuselaje utilizando un sistema de tornillos plásticos y tetones (varillas sólidas de fibra de carbono de 5[mm] de diámetro).

El receptor, el regulador de voltaje y la LiPo de alimentación de la radio están montados en la parte anterior del fuselaje. Las LiPos se ubican por delante de la cuaderna del tren de aterrizaje delantero, a fin de contribuir en el posicionamiento del centro de gravedad (CG). El servo de dirección va montado directamente en la cuaderna del tren delantero. El servo del acelerador del motor va montado lo más cerca posible del motor. Mientras más corto es el *link* que une al cuerno del servo con el cuerno de la superficie que maneja, menor será el juego que tendrá esa superficie y por lo tanto, mayor será la precisión del movimiento. Esta regla se aplica para todos los mandos del aeroplano.

En cada panel externo del ala va montado un servo encargado de un alerón. Éste se ancla dentro del ala mediante una fijación de madera de pino pegada a la plumavit. Sólo se asoma el cuerno del servo, ya que el cable va por dentro del ala.

En la cola van montados los 3 servos para manejar los dos timones y el elevador. Estos se fijan utilizando una montura de madera de pino pegada a sus respectivas superficies (fijas).

Todos los servos se encuentran a una distancia considerable del receptor (a excepción del control de la rueda delantera), por lo que es necesario hacer alargadores para los cables. El cable elegido es un AWG 22, para evitar atenuaciones en la señal y caídas de voltaje con el consecuente movimiento erróneo y la disminución de torque, respectivamente.

Los cables provenientes de la cola se rutean por uno de los boom's y luego entran también al ala. Todos los cables salen en el centro del intradós del panel central. A la hora de armar el modelo resulta muy engorroso conectar todos los cables uno a uno. Por ello se soldó un conector DB9 al cual llegan todas las señales y alimentación de los servos (Figura 4.2.9). De esta manera, resulta más fácil realizar la conexión del conjunto alas-cola con el fuselaje. En el conector DB9 se soldaron condensadores de 0.1[uF] en todas las alimentaciones a modo de filtros.

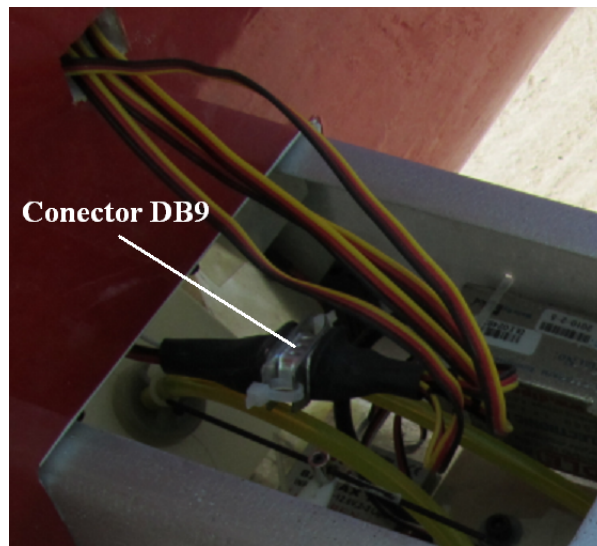


Figura 4.2.9: Detalle del conector DB9

Con el avión completamente ensamblado se puede pintar el fuselaje. Para la pintura del fuselaje se utilizó un spray epóxico de color gris. Este spray tiene la ventaja de que al ser epóxico se adhiere perfectamente al fuselaje y brinda una adecuada resistencia al combustible utilizado por el motor. El resto del avión (alas, cola y booms) es *monokoteado*. El monokote es una lámina plástica de color que en uno de sus lados está impregnada con un pegamento sensible al calor. El monokote ya dimensionado se coloca sobre la superficie a cubrir y se aplica calor con una plancha, el pegamento de la cara en contacto con la madera se activa y se adhiere a la madera. Es un proceso que requiere mucha práctica para lograr resultados aceptables pues es muy fácil que se formen burbujas o se

arrugue. Con la lámina pegada es necesario recortar los sobrantes y repetir el proceso hasta terminar de cubrir todo lo que se desea. Los colores elegidos para el monokote son rojo y blanco. De esta forma se logra que el avión tuviera los colores del logo de la Facultad. Con el avión completamente armado es necesario realizar la programación del radio control para lograr que todos los mandos funcionen correctamente. Finalmente, se ubica el centro de gravedad y se realizan los ajustes necesarios para que éste quede en la posición correcta (11[cm] del borde de ataque del ala). Fue necesario agregar 1.2[kg] de plomo en el morro del avión para lograr el correcto balanceo.

Como el motor es nuevo, es necesario realizarle un rodaje. El rodaje se realiza con el motor alimentado con una mezcla rica en aceite mineral y rodando a bajas revoluciones, 2500[RPM], por un lapso de 2 horas. Una imagen del avión terminado y a punto de realizar el primer vuelo se muestra a continuación en la Figura 4.2.10.



Figura 4.2.10: Avión terminado pronto a realizar el primer vuelo

4.2.4. Prueba del prototipo

La única forma de validar el diseño es realizando una prueba de vuelo. Los resultados obtenidos se discuten en el siguiente capítulo.

4.3. Integración de la electrónica en aeroplano

La integración de la electrónica de control en el modelo supone la instalación de la placa madre, el módulo IR y una nueva batería de LiPo. La posición de la electrónica de control se muestra en la Figura 4.2.2. La fijación de la placa madre al fuselaje se realizó utilizando tornillos y separadores de goma. Los separadores de goma sirven para que el PCB no esté en contacto directo con el fuselaje y también para reducir las vibraciones transmitidas desde el fuselaje a la placa madre. El sensor IR se fijó en la parte inferior central del fuselaje utilizando tornillos y tuercas. La LiPo que alimenta la electrónica se posicionó junto a las LiPos que alimentan el equipo de radio y el CDI del motor, en el morro del avión por delante de la cuaderna del tren de aterrizaje delantero.

4.3.1. Implementación del software

El software que ejecuta el microcontrolador es el encargado de realizar las lecturas de los sensores y a partir de éstas realizar el control sobre el aeroplano. Se han implementado dos programas: el primero se usa para realizar una toma de datos y enviarlos a tierra, mientras que el segundo realiza el control en tiempo real sobre el aeroplano. La estructura básica de ambos programas es muy similar, tal como puede apreciarse al comparar la Figura 4.3.1 y la Figura 4.3.2. El compilador y programador usados son los mismos que se utilizaron en la realización de las pruebas de los distintos módulos: PICC y PICKit. Al diseñar el software, los principales objetivos deben ser la confiabilidad y estabilidad, debido a que cualquier error en la ejecución del programa puede bloquear al microcontrolador y causar graves estragos si el UAV se encontrase en modo autónomo. La estructura del software, tanto de control como para la toma de datos, está definida en 3 bloques: inicialización, rutina principal e interrupciones.

Inicialización

En la primera parte de la inicialización se configuran y activan todos los módulos del microcontrolador que serán utilizados, por ejemplo los puertos seriales. A continuación se inicializan todas las variables de estado que permiten el correcto funcionamiento de la rutina principal. En estas variables se incluyen 11 parámetros de calibración, necesarios para obtener lecturas reales del sensor de presión absoluta. Tras la inicialización de las variables de estado activan las interrupciones.

Rutina principal

La rutina principal se comporta como el Kernel del programa. Está basado en interrupciones de tiempo. De esta forma, se logra ejecutar bloques de tareas en momentos determinados y de manera sincronizada, a modo de sistema operativo. La interrupción del *Timer2* sucede cada 20[ms] y en ella se incrementan diversos contadores relacionados con las distintas tareas que se quiere realizar. La rutina principal revisa constantemente el estado de estas variables. En el caso de que el contador de una de las tareas alcance el valor deseado, se ejecuta la rutina correspondiente a ese contador.

Interrupciones

Se habilitaron un total de 3 interrupciones: interrupción de recepción de datos en el puerto serial #2, interrupción del *Timer2* y la interrupción generada por el *Timer3*.

La interrupción asociada a la recepción de datos por el puerto serial #2 se gatilla cada vez que se recibe un nuevo caracter. Estos caracteres son los enviados por el GPS y es la sentencia NMEA RMC. La rutina de interrupción detecta el carácter que da inicio a la sentencia (“\$”) y reinicia un contador. Los caracteres recibidos son guardados en un arreglo hasta que se detecta el caracter que indica el fin de la sentencia (“*”). El arreglo es entonces copiado a un nuevo arreglo desde el cual se realizará la correspondiente interpretación.

La interrupción asociada al *Timer2*, como se explicó anteriormente, se ejecuta cada 20[ms] y en ella simplemente se incrementan contadores.

La interrupción del *Timer3* es la que generará el tren de pulsos enviado al módulo conmuta/decod (*Clock* en la Figura 3.1.11). El período de cada pulso es determinado por la función encargada de realizar el control de estabilidad y se encuentra guardado en un arreglo global. Cada vez que se ejecuta la interrupción, se incrementa el índice del arreglo en el que se encuentran los

períodos de los pulsos. Este período determina el momento en que se ejecutará nuevamente la interrupción.

Los programas utilizados en las pruebas de las distintas funcionalidades de la electrónica sirven como base para la programación de las distintas rutinas, pues cuentan con las funciones básicas de interfaz con los sensores. La lectura de los sensores IR se hace utilizando una función que lee el canal ADC que se le entrega como parámetro y luego retorna el resultado. De la lectura del GPS se encarga la rutina de interrupción serial, ella guarda toda la sentencia recibida en una cadena de caracteres. Cuando se envían los datos se realiza el parseo, que es otra función, y que almacena el resultado en variables globales. La lectura del sensor de presión absoluta se ejecuta también en forma de función, simplemente se llama a la función *lecturai2c(registro)* que recibe como parámetro el registro a leer (presión o temperatura) y retorna el resultado de la lectura. La rutina de reset del XBee depende de la cantidad de datos enviados en un lapso determinado de tiempo, al realizar una prueba se determinó que el XBee interrumpía la transmisión de información transcurridos 6 minutos desde que comenzaba la ejecución del programa. Este lapso de tiempo depende del volumen de datos enviados. Con el reset es posible evitar esta interrupción. Es necesario tener en consideración que el XBee se calienta mucho y por lo tanto, no es recomendable hacerlo funcionar a una tasa de transmisión muy elevada durante largos períodos de tiempo.

Comandos seriales

Como fue explicado con anterioridad, la comunicación entre el microcontrolador y el exterior se lleva a cabo a través del puerto serial #1. La información puede ser enviada mediante el *link* RF o simplemente utilizando una conexión directa mediante el conversor serial-USB. El protocolo de comunicación es el utilizado por Parra en [32]. Este protocolo se basa en la utilización de sentencias que comienzan con el caracter “[“ y terminan con el caracter “]” y dentro de la sentencia los campos de datos son separados por el caracter “|”. El primer campo indica el tipo de mensaje que se envía y los siguientes campos son los datos. La ventaja de este protocolo es la fácil lectura y la simple decodificación. Para realizar la decodificación de los datos recibidos, se escribió un programa en MATLAB® que identifica y separa los comandos.

4.3.1.1. Software para la toma de datos

El software para la toma de datos únicamente envía las lecturas de los sensores a tierra mediante el *link* RF. Este programa no implementa la interrupción del *Timer3* ni la función que realiza el control de estabilidad. Con este programa se busca comprobar el correcto funcionamiento de los sensores que serán utilizados para realizar el control de estabilidad en vuelo y obtener lecturas reales de estos, para así diseñar una estrategia de control de acuerdo a lo observado.

Las rutinas y el momento en que se ejecutan se detallan en la siguiente tabla:

Tabla 4.3.1: Período ejecución rutinas para la toma de datos en modo RC

Rutina	Período [ms]
Lectura sensores IR	20
Envío lectura sensores IR	60
Lectura GPS y presión absoluta	980
Envío datos GPS y presión absoluta	1000
Reset XBee	280000 (~ 4,6 minutos)

El diagrama de flujos mostrado en la Figura 4.3.1 muestra el código implementado para realizar la toma de datos de prueba en modo RC.

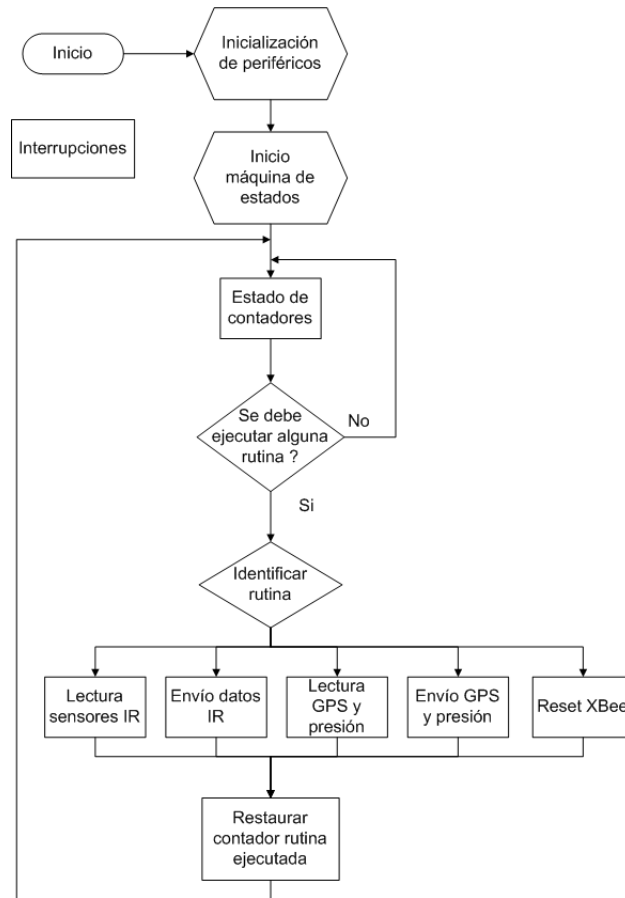


Figura 4.3.1: Diagrama de flujos del programa principal para la toma de datos en modo RC

4.3.1.2. Software de control para la operación en modo autónomo

A diferencia del software desarrollado para la toma de datos, en este caso sí se implementa la interrupción del *Timer3* y la función que realiza el control de estabilidad. La estabilización se realizará puramente en función de la lectura de los sensores IR. A partir de estas lecturas es posible obtener, mediante las ecuaciones 3.2.4 y 3.2.5, el cabeceo y alabeo del aeroplano. Cada vez que se ejecuta la rutina de la lectura de los sensores IR, se realiza la conversión que permite obtener el cabeceo y alabeo. A continuación se le entregan estos parámetros a la función *control(cabeceo,alabeo)*. Para realizar la estabilización es necesario manejar los alerones y el elevador, esto implica determinar los pulsos para 3 servos (2 alerones y 1 elevador). La función *control(cabeceo,alabeo)* implementa un controlador proporcional, cuyo objetivo es mantener la horizontalidad del aeroplano. La referencia a seguir para el controlador será entonces 0, tanto para

el cabeceo como para el alabeo. El ancho inicial del pulso para cada servo a manipular corresponde a su posición neutral. A partir de esta posición, y dependiendo del error, el controlador proporcional aumenta o disminuye el ancho del pulso. Las ganancias k_{cabec} y k_{alab} son determinadas analizando los datos obtenidos en la toma de datos y si es necesario se sintonizará utilizando el método de prueba y error.

El controlador proporcional es utilizado por su simplicidad y su rápida respuesta. La principal desventaja de este tipo de controlador es la imposibilidad de corregir algunos errores en régimen permanente. Si bien el vuelo de un avión en condiciones favorables es estable, basta que existan pequeñas corrientes de aire que lo desestabilicen, esta es la realidad en la mayoría de los casos. Por lo tanto, el aeroplano sufrirá perturbaciones constantemente. La imposibilidad de corregir errores permanentes frente a la rapidez de actuar del controlador proporcional convierten a ésta última característica en más importante.

Durante el vuelo autónomo sólo se envía un tipo de sentencia a tierra. Esta sentencia indica en su primer campo de datos el modo en que se encuentra el UAV: modo RC o modo autónomo. En los dos siguientes campos se indica el cabeceo y el alabeo.

La siguiente tabla muestra las rutinas que se ejecutan en modo autónomo y el período de su ejecución:

Tabla 4.3.2: Período ejecución rutinas en el software de control en modo autónomo

Rutina	Período [ms]
Detección estado RC/autónomo	Inicio del Loop principal
Lectura sensores IR y control	20
Envío cabeceo y alabeo	160
Reset XBee	280000 (~ 4,6 minutos)

El siguiente diagrama de flujos (Figura 4.3.2) muestra el código implementado para realizar un vuelo en modo autónomo:

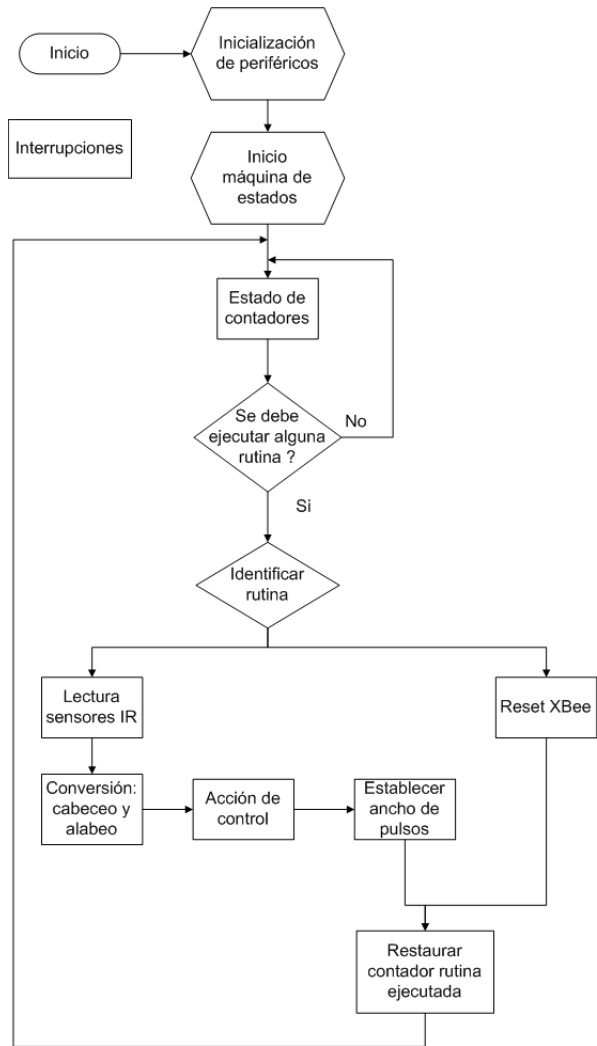


Figura 4.3.2: Diagrama de flujos del programa principal para el vuelo en modo autónomo

Capítulo 5

Resultados y análisis

Los resultados obtenidos del trabajo se dividen en 4 secciones:

- Pruebas realizadas al aeroplano.
- Sensores y electrónica.
- Adquisición de datos durante vuelos de prueba en modo RC.
- Pruebas del sistema de control.

5.1. Pruebas realizadas al aeroplano

La prueba de vuelo se realizó en el Club de Aerodelismo RC-Carén ubicado en el complejo de la Universidad de Chile: Laguna Carén.

Todos los vuelos de prueba fueron realizados sin la electrónica de automatización en el aeroplano, ésta se instalará sólo una vez completas las diferentes pruebas al aeroplano. Con esto se busca reducir la posible pérdida total del prototipo en el supuesto de un accidente.

La siguiente figura muestra una secuencia de aproximación a la pista de aterrizaje:



Figura 5.1.1: Aproximación a la pista de aterrizaje

El primero reveló algunos detalles técnicos que fue necesario corregir. Estos detalles están ligados netamente a componentes o piezas del aeroplano y no al diseño aerodinámico.

1. Tren de aterrizaje trasero muy blando: El tren de aterrizaje fue construido en aluminio de 3[mm] de espesor. El tipo de aluminio utilizado en su construcción es común y sin aleaciones que aumenten su rigidez. El material ideal es duraluminio (aluminio con aleaciones que aumentan su dureza, rigidez y son más fáciles de mecanizar) pero es muy difícil de conseguir en forma de láminas. Los trenes de aterrizaje que se pueden encontrar en tiendas especializadas no eran de las dimensiones requeridas, pues están generalmente diseñados para aeroplanos con motores en configuración tractora (la hélice se separa del suelo al despegar, no así en una configuración pusher) por lo que se decidió construir uno a medida. El tren se comportó de manera adecuada en tierra y durante el despegue, pero al momento del aterrizaje el aluminio mostró su debilidad doblándose hacia afuera, acercando peligrosamente la hélice al suelo. Para el siguiente vuelo de prueba se acomodó manualmente pero en el siguiente aterrizaje se repitió el problema. La solución final consistió en agregar dos tirantes de alambre de acero galvanizado. Los tirantes salen de una posición muy cercana a las ruedas y se anclan en una posición central al brazo opuesto del tren. La configuración en “X” de los tirantes distribuye mejor las fuerzas evitando que el tren se doble pero sin limitar la flexibilidad del conjunto (los tirantes no están tensos). En posteriores vuelos esta solución probó su efectividad.

2. Tren de aterrizaje delantero muy flexible y débil: Inicialmente el tren fue hecho en acero de 4[mm] de diámetro. El tren de aterrizaje delantero es esencial en tierra, pues es el encargado de dotar al aeroplano de dirección durante el carreteo, despegue y aterrizaje. La elevada masa en el morro del aeroplano (más de 1[kg] en plomo para lograr el adecuado CG) y la gran altura a la que se encuentra el fuselaje respecto al suelo requerían de una varilla más rígida. La varilla de 4[mm] se flecta en todas direcciones con mucha facilidad, imposibilitando el control del aeroplano en tierra. En el aterrizaje simplemente se dobló hacia atrás, terminando el morro del fuselaje apoyado sobre la pista de aterrizaje. La solución consistió en aumentar significativamente el diámetro de la varilla a 6[mm] hecha de acero templado para resortes. De esta forma se solucionó el problema.
3. Estanque de combustible instalado al revés: Se utilizó la técnica de construcción en aeroplanos con motores en configuración tractora. En esta configuración el estanque de combustible va montado a continuación del motor. La toma de combustible para el motor va en la parte posterior del estanque, o sea, se aleja del motor. Al momento del despegue la bencina se ve sometida a la aceleración y fuerza de gravedad y por lo tanto se desplaza hacia la toma de combustible evitando que el motor se quede sin suministro de combustible. Para el prototipo simplemente se rotó esta configuración, lo que terminó con la toma de combustible en la parte posterior del estanque (alejada del motor). Este error se tradujo en que al momento del despegue la toma de combustible se quedara en seco y el motor se detuviese. En una configuración pusher, la toma de combustible debe estar en la cara del estanque más cercana al motor y no la más alejada. Se acortó la manguera de la toma para que esta quedara en la posición adecuada (cercana al motor).
4. Toma de presión estática del estanque de combustible mal ubicada dentro del avión: La toma de presión estática sirve para evitar que en el estanque se forme un vacío cuando la bencina es succionada por el motor. También sirve como indicador de un estanque lleno de combustible, al llenarse la única salida que tiene el combustible es esta toma. Inicialmente esta toma se ubicó al interior del fuselaje, no pudiendo determinar cuando el estanque estaba lleno y con la posibilidad de derramar combustible en el interior del fuselaje. La solución consistió en alargar la manguera de la toma para que esta quedara fuera del fuselaje.

5. Funcionamiento irregular y erróneo en los servos de la cola: El problema más grave y por el cual fue necesario cancelar los vuelos se presentó en fechas posteriores al primer vuelo. Los servos digitales utilizados en la cola tienen la posibilidad, mediante un programador externo, de ajustar ciertos parámetros de su funcionamiento sin utilizar conexiones adicionales al servo. Esto implica que la línea de señal de control del servo es bidireccional. Para configuraciones especiales, como cuando se utilizan cables muy largos y una “Y” para manejar dos servos (timones), se puede dar el caso que los servos entren en un modo indefinido y “hablen” en las líneas de señal de control provocando un mal funcionamiento. Para evitar este problema se diseñó y construyó un “Buffer” que se instaló en la cola. Se utilizó un “AND” para cada señal de control. Además de funcionar como un amplificador de la señal que llega a cada servo sirve para aislar (utilizando lógica separada) unas de otras las líneas de señal que antes eran separadas por la “Y”. La sencillez del circuito no requiere de complejos ruteos y permite que para su construcción se utilice una placa perforada. La siguiente figura muestra la configuración de los ANDs y el circuito terminado.

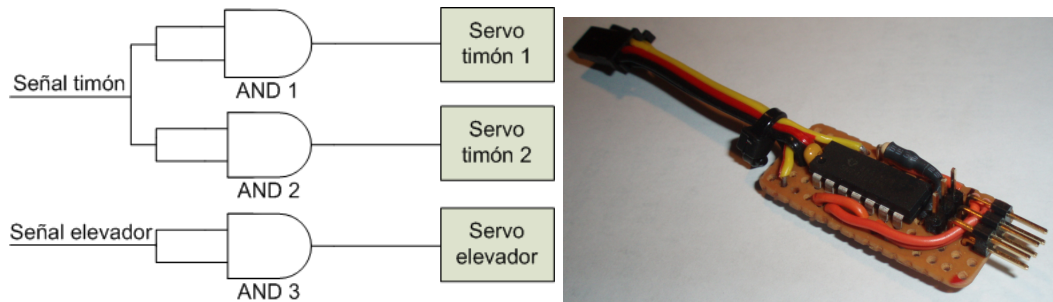


Figura 5.1.2: Circuito del Buffer

Las siguientes pruebas de vuelo consistieron en despegar y realizar una serie de figuras y así validar tanto la parte estructural del diseño como el diseño en sí. Estas figuras incluyeron: rolls, loops y caídas de ala, entre otros. Las características del vuelo son las esperadas según el diseño:

- Vuelo estable en amplio rango de velocidades, es posible soltar los mandos y el avión seguirá su curso.
- Despegue y aterrizaje corto.
- Potencia suficiente del motor, lo que implica que se cuenta con la capacidad de incluir nuevos sensores u otras cargas.

- Facilidad de manejo, al ser estable, se evita que los movimientos sean bruscos o inesperados.
- La estructura soportó todas las maniobras sin observarse flexiones o vibraciones no deseadas.
- La posterior revisión en tierra no reveló ningún problema estructural.

Una vez que se realizaron, con éxito, todas las pruebas de vuelo, se procedió a instalar la electrónica de automatización en el aeroplano.

5.2. Sensores y electrónica

El diseño y construcción de los PCBs se realizó sin complicaciones. Las experiencias anteriores en diseño y construcción de PCBs facilitaron en gran medida todas las tareas referentes a este ámbito, siendo necesaria la construcción de un único prototipo de todas las placas. Fue necesario aprender a utilizar la LPKF, debido a que las placas se construyeron en una época en que no se encontraba el encargado de operar la máquina. Conocimientos adquiridos en ocasiones previas del manejo de máquinas CNC facilitaron esta tarea.

Las pruebas de laboratorio realizadas a los distintos módulos tuvieron un resultado exitoso, revelando sólo el sensor de presión absoluta algunos problemas. Para obtener lecturas válidas, es imprescindible realizar una gran cantidad de cálculos que incluyen los parámetros de calibración y las lecturas de los datos de presión y temperatura no compensados. Si bien el cálculo está explicado en el *datasheet* del integrado [17], es difícil de implementar en el microcontrolador y los datos finalmente obtenidos no son fieles a la realidad. Se buscó información en Internet sobre como superar los problemas y se observó que, efectivamente, la implementación de este sensor es compleja. Observando los parámetros de calibración obtenidos y comparándolos con los del ejemplo expuesto en el *datasheet* del integrado se nota una diferencia importante en 2 de los 11 parámetros. Cambiando estos 2 parámetros entregados por el sensor a los 2 del ejemplo se obtienen lecturas más cercanas a la temperatura ambiente esperada, no así de la presión absoluta.

El módulo GPS cuenta con una batería interna para que no se borre la configuración. Sin embargo, la configuración se borra cuando el módulo permanece desenergizado largos períodos de tiempo. Este error es difícil de detectar y obliga a tener siempre una manera auxiliar de configurar el módulo sin utilizar el microcontrolador.

La integración de la electrónica de automatización en el aeroplano también resultó problemática. El montaje de la placa madre mediante tornillos no fue el adecuado. Después de la primera prueba con la electrónica montada, se descartó este sistema. Las altas vibraciones y extremas fuerzas G a la que se ve sometido el modelo durante el vuelo hicieron que se salieran el módulo de alimentación y presión de sus sockets en la placa madre. Este problema fue resuelto montando la placa madre sobre tiras de goma-espuma fijas al fuselaje. De esta forma se logra que la placa madre quede suspendida sobre la goma-espuma, que absorbe parte de las vibraciones. Los módulos de presión y alimentación se fijaron utilizando pegamento epóxico para prevenir que se soltaran nuevamente.

5.3. Adquisición de datos durante vuelos de prueba en modo RC

Los vuelos de prueba con la electrónica montada consistieron en realizar tomas de datos de los sensores IR y GPS mientras el avión ejecutaba distintas piruetas en el aire.

El *link* RF funciona correctamente, pero es necesario estar apuntando la antena receptora en dirección al aeroplano para no perder la recepción. Si no se apunta constantemente al avión, la cantidad de datos recibidos decrece notablemente. Este problema se debe a la manera de radiar de las antenas y la posición de la antena transmisora en el aeroplano. Esto sólo supone un problema para la etapa de captura de grandes cantidades de datos, en este caso a una tasa de $18[\frac{\text{paquetes}}{\text{s}}]$. Una vez superada la etapa de captura, es posible disminuir la frecuencia de envío de datos, pues estos sólo sirven para informar del estado del UAV ($2[\frac{\text{paquetes}}{\text{s}}]$).

El sensor IR probó tener gran resolución y bajo ruido. Esta calidad de información brinda la posibilidad de realizar el control de estabilidad horizontal basándose sólo en la lectura de estos sensores. Sin embargo, no es posible determinar la altura de vuelo, ya que para esto es necesario el sensor de presión absoluta. En caso de que los problemas no se puedan superar con este sensor, sería posible utilizar la lectura de la altura del GPS como referencia, pero realizando un filtrado para evitar cambios bruscos.

El siguiente es un ejemplo de la lectura obtenida por los sensores IR cuando el aeroplano ejecuta una pirueta denominada “8 horizontal”.

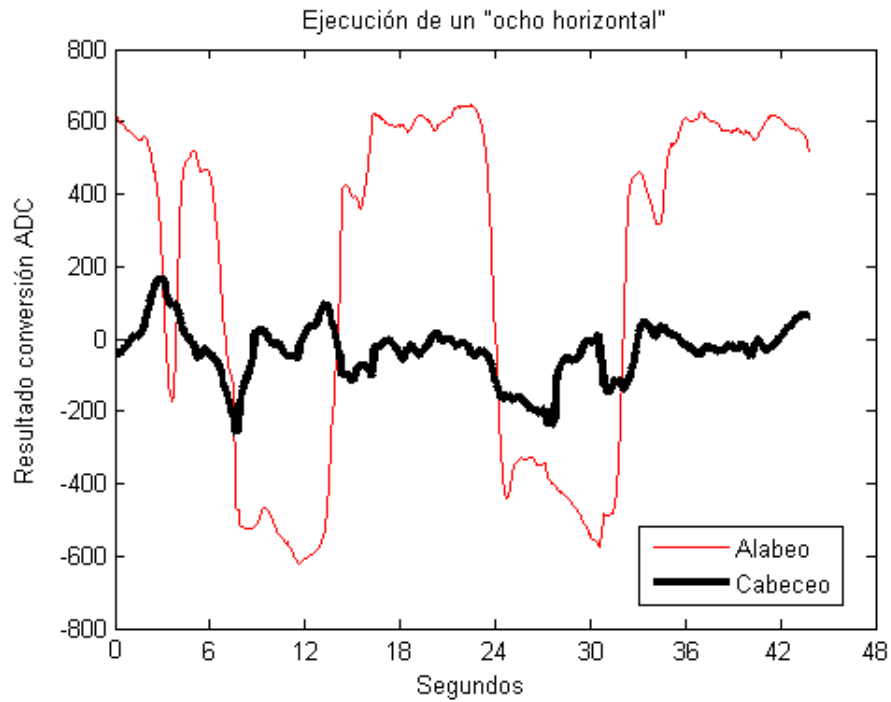


Figura 5.3.1: Ocho horizontal IR

En la Figura 5.3.1 se aprecia claramente, observando la curva “Alabeo”, el momento en que el aeroplano ejecuta un círculo hacia la izquierda (medición 90 – 220), un círculo hacia la derecha (medición 230 – 390) y como repite la pirueta una vez más. La curva “Cabeceo” muestra como desde el momento en que se comenzó con la pirueta (medición 90) el aeroplano mantuvo su ángulo de cabeceo casi constante, lo que es coherente con la horizontalidad de la pirueta ejecutada (la altura debe ser constante). Ambas mediciones se encuentran centradas en cero, que es el punto en que el aeroplano se encuentra completamente en el plano horizontal.

Los datos enviados por el GPS se muestran en la siguiente figura.

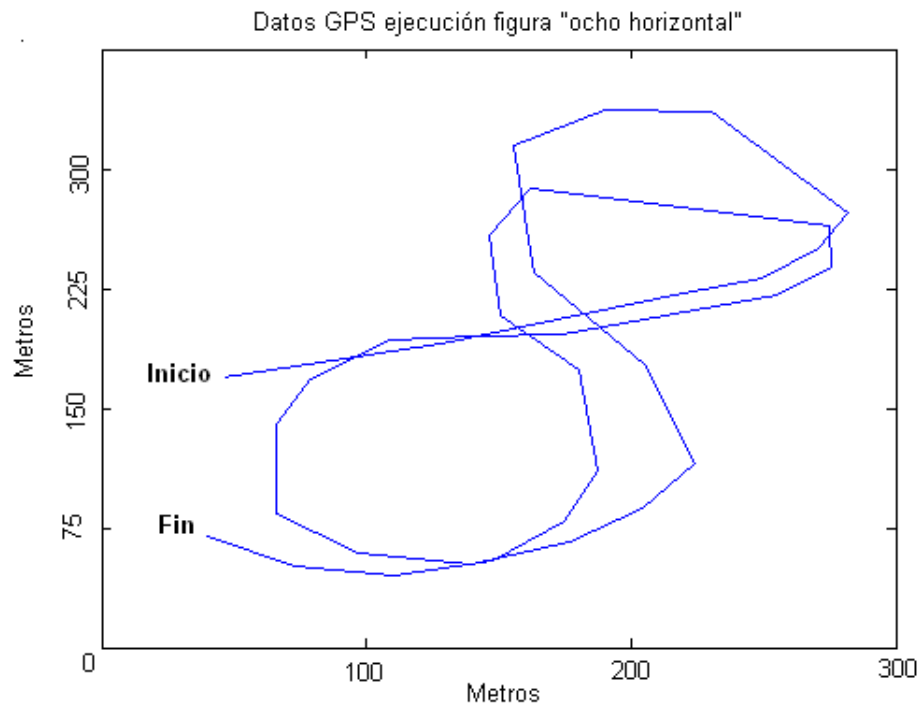


Figura 5.3.2: Ocho horizontal GPS

Se observa la confiabilidad y la calidad de este sensor, pues los dos “8” se aprecian claramente. Es importante destacar que el envío de los datos del GPS se realiza cada aproximadamente 1[s] (refresco de 1[Hz]), esto explica las discontinuidades presentes en algunos sectores del “8”. En caso de ser necesario, el GPS puede realizar un refresco de la posición con una frecuencia de hasta 5[Hz].

5.4. Pruebas del sistema de control

Las pruebas del sistema de control consisten en realizar conmutaciones entre el modo RC y el modo autónomo durante la ejecución de distintas piruetas. Como medida de seguridad, todas las conmutaciones entre estados se realizaron a alturas elevadas. De esta forma, si se observa un mal funcionamiento del controlador, se puede retornar al modo RC a tiempo. Al realizar la conmutación de modo RC a modo autónomo, el microcontrolador se hace cargo del control de los servos de los alerones y elevador; y buscará estabilizar al aeroplano en base a las lecturas del sensor IR. Una sencilla prueba en tierra, simulando distintas posiciones de vuelo del aeroplano, revela si el conmutador funciona correctamente y si el controlador mueve los servos en las direcciones esperadas.

Las ganancias fueron ajustadas en tierra simulando el vuelo del aeroplano, esto se puede hacer dada la experiencia con que se cuenta, ya que se conoce la posición de los mandos del radiocontrol para corregir los ángulos de cabeceo y alabeo bajo distintas condiciones. Así, es posible comparar la corrección que haría un piloto con la corrección que realiza el controlador y de esta forma, ajustar las ganancias.

Las pruebas se realizaron en un día con condiciones favorables de visibilidad y cielo muy despejado. Estas condiciones facilitan la detección del diferencial de temperatura entre la atmósfera y la superficie terrestre. Las primeras pruebas consisten simplemente en realizar conmutaciones para observar repentinos cambios en la actitud del aeroplano. Al no observar cambios se prueba el correcto funcionamiento y trimado del controlador. También se comprobó la correcta recepción de datos mediante el *link* RF.

Primero se probó la estabilización del alabeo: se maniobró el aeroplano hasta dejarlo levemente inclinado y luego se realizó la conmutación. La estabilización se produce en un breve lapso de tiempo. Luego, se aumentó gradualmente el ángulo de alabeo para ver la respuesta del controlador. En todos los casos el resultado fue el esperado: una rápida estabilización. El mismo esquema de pruebas se realizó con el cabeceo, obteniendo similares resultados. Una vez comprobado el funcionamiento del controlador para el alabeo y el cabeceo se hicieron pruebas con el aeroplano desestabilizado en ambos ejes. Se probó incluso la respuesta cuando el aeroplano se encuentra volando invertido, obteniendo como resultado la mitad de un roll para devolver al UAV a la horizontalidad con el plano terrestre en posición no-invertida.

Durante el vuelo el microcontrolador envía 3 variables en un mismo paquete: cabeceo, alabeo y estado de conmutación. La siguiente figura muestra los datos obtenidos durante un vuelo de prueba.

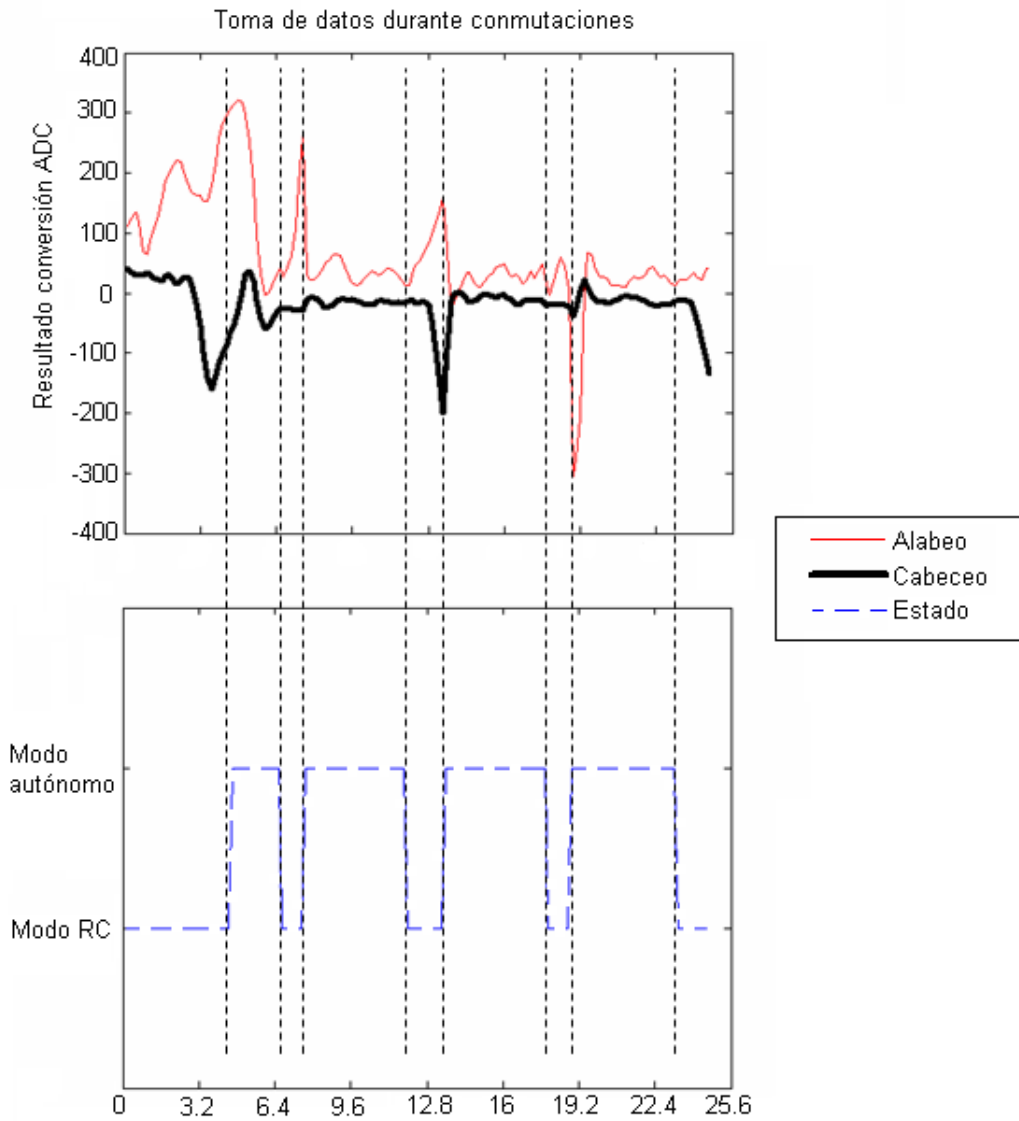


Figura 5.4.1: Resultado de pruebas en modo autónomo

Cuando “estado” está arriba, el UAV se encuentra en modo autónomo y cuando está abajo, en modo RC. Los momentos en los que se realizó la conmutación son claramente instantes en que el aeroplano se encontraba altamente desestabilizado. Se observa la rápida acción de control y la alta efectividad: la sobreoscilación es pequeña al igual que los errores permanentes. Las oscilaciones en torno a la referencia se pueden deber a las constantes perturbaciones a la que se ve afectado el UAV y al controlador tipo P. Estas oscilaciones no se aprecian desde tierra, lo que hace suponer que el

vuelo en modo autónomo es muy estable, incluso mejor que el vuelo en modo RC. Las ganancias no fueron ajustadas nuevamente después de la sintonización en tierra.

No se realizaron pruebas en vuelo del sensor de presión diferencial. Sólo falta instalar un tubo de Pitot en el fuselaje, unirlo al sensor mediante mangueras de silicona y realizar una calibración para que éste comience a funcionar.

Un problema importante en el desarrollo e implementación del software fue el compilador. El microcontrolador utilizado es relativamente nuevo y el compilador tiene muchos problemas para implementar correctamente algunas funciones. Fue necesario cambiar la estructura del programa y versión de compilador varias veces a fin de lograr que el programa se ejecutara correctamente. Se intentó implementar un sistema operativo en el microcontrolador utilizando las librerías incluidas con el compilador, pero el microcontrolador se reseteaba constantemente. Este fue el motivo por el cual se terminó implementando un sistema secuenciado basado en contadores e interrupciones de tiempo, el cual no tiene todas las funcionalidades del sistema operativo incluido con el compilador (PICC® 4.084) pero realiza las tareas necesarias. Realizando una investigación para descubrir el origen del problema se encontró que el compilador tiene un problema con el uso de punteros, este problema aun no ha sido resuelto por la empresa desarrolladora del compilador. El mal funcionamiento del sensor de presión absoluta puede estar relacionado con este problema.

Capítulo 6

Conclusiones y trabajo futuro

6.1. Conclusiones

En este trabajo se presentaron todas las etapas comprendidas en el proceso de diseño y construcción de un vehículo aéreo no tripulado. Se logró con éxito construir una plataforma aérea capaz de transportar la electrónica necesaria para su automatización y otros sensores o cargas adicionales. El aspecto más destacable del aeroplano es que se trata de un diseño y construcción que partió de cero, no se adquirió un aeroplano para ser modificado ni se construyó uno en base a planos de otro aeroplano. El aeroplano construido cumplió con las características de vuelo esperadas según el diseño y los objetivos planteados.

El objetivo referente al diseño y construcción de la electrónica de automatización fue cumplido exitosamente. No fue necesario realizar reparaciones en las placas o hacerlas nuevamente, pues el diseño original funcionó como era esperado en el primer intento gracias al cuidadoso diseño y experiencia. Las tomas de datos validaron el correcto funcionamiento de los sensores, salvo el de presión absoluta que entrega lecturas anormales. La toma de datos en modo RC reveló por qué el sensor IR es ampliamente utilizado en UAVs. Su fácil implementación y reducido costo lo convierten en el sensor principal en las aplicaciones de estabilización más sencillas. La excelente calidad de los datos permitió realizar el control de estabilización basándose sólo en las lecturas del sensor IR. El resultado del vuelo en modo autónomo reveló la simplicidad con la que se puede implementar un control de estabilización con el controlador más sencillo: proporcional. Si bien el controlador es

simple, los resultados mostraron que era suficiente y que cumplía con los objetivos de las pruebas. Incluso, situando al aeroplano en posiciones extremas (vuelo invertido) al realizar la conmutación a modo autónomo el controlador logró su objetivo y devolvió al UAV a un vuelo no-invertido.

En caso de que se desee retomar el proyecto, se deja una amplia documentación de respaldo. Los códigos, tanto de programas finales y de pruebas, han sido comentados extensamente para brindarle al interesado una lectura más accesible.

Finalmente, cabe destacar que el proyecto realizado es el primero de esta naturaleza en ser implementado enteramente en la Facultad.

Los objetivos relacionados con la difusión en el Departamento de Ingeniería Eléctrica también se cumplieron. Se realizó una charla demostrativa del trabajo a los alumnos de plan común, en el marco de la muestra de algunos proyectos que se realizan en el departamento.

6.2. Trabajo futuro

La electrónica de automatización deja implementada una instrumentación muy completa, que permite, mediante la programación de nuevas rutinas de control, realizar diversas misiones en modo autónomo. Sin embargo, el compilador es un problema que limita las capacidades del microcontrolador. Por este motivo, puede ser recomendable migrar el código a otro compilador, como por ejemplo MPLAB®. MPLAB® es un compilador desarrollado por Microchip™, que permite realizar programación en alto y bajo nivel sin la abstracción presente en PICC®. También se pueden esperar versiones futuras de PICC® en el que arreglen los actuales problemas. Sería de gran utilidad poder implementar el sistema operativo provisto por este compilador. En el caso del sensor de presión absoluta, también es necesario realizar un extenso debugueo para descubrir la falla que presenta. Existen alternativas menos sofisticadas para determinar la altura de vuelo, como es utilizar un sensor de presión analógico. Con éste no se tienen todas las ventajas del sensor Bosch™, pero su implementación es significativamente más sencilla.

Para el envío de datos se pueden cambiar las antenas de los módulos RF por antenas omnidireccionales, y de esta forma no se requiere estar apuntando la antena de la estación en tierra constantemente hacia el aeroplano. También se puede implementar un protocolo de confirmación de recepción de datos, para asegurar una comunicación libre de errores.

El aeroplano deja abierta las posibilidades de montar nuevos sensores. Estos sensores pueden

ser utilizados para mejorar el control en modo autónomo o para realizar toma de datos, como fotografías o temperaturas. Para mejorar la estimación de posición el aeroplano, se puede añadir una IMU y un nuevo par de sensores IR en el eje Z. La integración de una IMU con los sensores IR es una solución interesante para evitar los problemas del error asociado a la lectura de los giróscopos.

Para trabajos futuros se deja a disposición del estudiantado una plataforma de desarrollo de sistemas de control y posicionamiento.

Bibliografía

- [1] Armada International 5/2001, Complete Guide to Drones and Sensors.
- [2] Shun-Wen Cheng. Rapid Deployment UAV. Aerospace Conference, 2008 IEEE.
- [3] Ollero Anibal, Merino Luís. Control and perception techniques for aerial robotics. Annual Reviews in Control 28 (2004) 167–178
- [4] Ollero A., Martínez-de-Dios J.R., Merino L.. Unmanned aerial vehicles as tools for forest-fire fighting. V Intl Congress on Forest Fire Research – V ICFRR 2006. Nov. 2006.
- [5] Zhou Guoqing, Zang Deyan. Civil UAV System for Earth Observation. Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE Internacional.
- [6] Brown Alison. Test Results of a GPS Inertial Navigation System using a Low Cost MEMS IMU. Proceedings of 11th Annual Saint Petersburg International Conference on Integrated Navigation System Saint Petersburg, Russia, May 2004.
- [7] Fu Xu, Zhou Zhaoying, Xiong Wei, Guo Qi. MEMS-Based Low-Cost Flight Control System for Small UAVs. Tsinghua Science and Technology ISSN 1007-0214 07/25 pp614-618 Volume 13, Number 5, October 2008.
- [8] Jong-Hyuk Kim, Wishart Stuart, Sukkarieh Salah. Real-time Navigation, Guidance, and Control of a UAV using Low-cost Sensors. FSR , Vol. 24Springer (2003) , p. 299-309.
- [9] Scott M. Ettinger, Nechyba Michael C., Ifju Peter G., Waszak Martin. Vision-Guided Flight Stability and Control for Micro Air Vehicles. Proc. IEEE Int. Conference on Intelligent Robots and Systems, October, 2002

- [10] Moreno Leonardo. Diseño e implementación de vehículo autobalanceado sobre dos ruedas. Memoria para optar al título de Ingeniero Civil Electricista. Universidad de Chile, 2009. xiii, 110 h. : il.
- [11] Taylor B., Bil C., Watkins S.. Horizon Sensing Attitude Stabilisation: A VMC Autopilot. Presented to the 18th International UAV Systems Conference, Bristol, UK, 2003.
- [12] <http://www.ciga.cl/>
Fabricación de circuitos impresos, Ciga, 2010.
- [13] Abusleme Hoffman Angel Christian. Control Difuso de Vehículo Volador no Tripulado. Tesis ingeniero civil electricista. Universidad Católica de Chile, 2000.
- [14] Hatcher Norman M., Germann Nomran M.. Jr. NASA Technical Note D-1005. Washington 1962.
- [15] <http://www.nmea.org/>
The National Marine Electronics Association (NMEA), 2010.
- [16] <http://www.sparkfun.com/categories/17>
Sparkfun Electronics, 2010.
- [17] Bosch. Datasheet BMP085 Digital pressure sensor. Bosch, 2009.
- [18] Digi International. Xbee-PRO 868 OEM RF Module Operation, Configuration and Appendices. Digi International, 2008.
- [19] Texas Instruments. Datasheet TPS76350. Low-Power 150-mA Low-Dropout Linear Regulators. Texas Instruments, 2004.
- [20] Texas Instruments. Datasheet CD4017B, CD4022B Types. CMOS Counter/Divider. Texas Instruments, 2004.
- [21] http://paparazzi.enac.fr/wiki/Main_Page
The Paparazzi Project, 2010.
- [22] <http://www.vuelosdebautismo.com.ar/exposicion-aerea/vuelo-en-cessna.html> ,
<http://www.defenseindustrydaily.com/Dutch-to-Rent-Israeli-UAVs-for-Afghanistan-05254/>
Imágenes de configuraciones tractoras y pusher en distintos aeroplanos, 2010.

- [23] <http://www.worldofkrauss.com/>
Airfoil Investigation Database, 2010.
- [24] Long Di, Haiyang Chao, YangQuan Chen. Center for Self-Organizing and Intelligent Systems(CSOIS) Dept. of Electrical and Computer Engineering Utah State University, Logan, UT, USA, 2010.
- [25] Egan G.K., Taylor B.. The Use of Infrared Sensors for Absolute Attitude Determination of Unmanned Aerial Vehicles. Department of Electrical & Computer Systems Engineering Monash University 3800 Melbourne, Australia, 2006.
- [26] Texas Instruments. Datasheet PTR08060W, 6-A, 4.5-V to 14-V Input, Non-Isolated, Adjustable Wide-Output, Switching Regulator. Texas Instruments, 2009.
- [27] Texas Instruments. Datasheet TLV1117, Adjustable and Fixed Low-Dropout Voltage Regulator. Texas Instruments, 2008.
- [28] Freescale Semiconductor. Technical Data MPX5010, MPXV5010G Series. Freescale Semiconductor, 2007.
- [29] Microchip. Datasheet PIC24FJ128GA010 Family. Microchip, 2007.
- [30] FTDI. Datasheet ft232r usb uart i.c. Technical report, FTDI, 2005.
- [31] ETEK Navigation, Inc. MTK NMEA Packet User Manual, Revision 0.3. MediaTek Inc, 2006.
- [32] Parra Tsunekawa Sebastián Isao. Diseño e Implementación de Sistemas de Control para Robots Bípedos. Memoria para optar al título de Ingeniero Civil Electricista. Universidad de Chile, 2008.

Anexos

Esquemáticos y placas módulos

Módulo alimentación

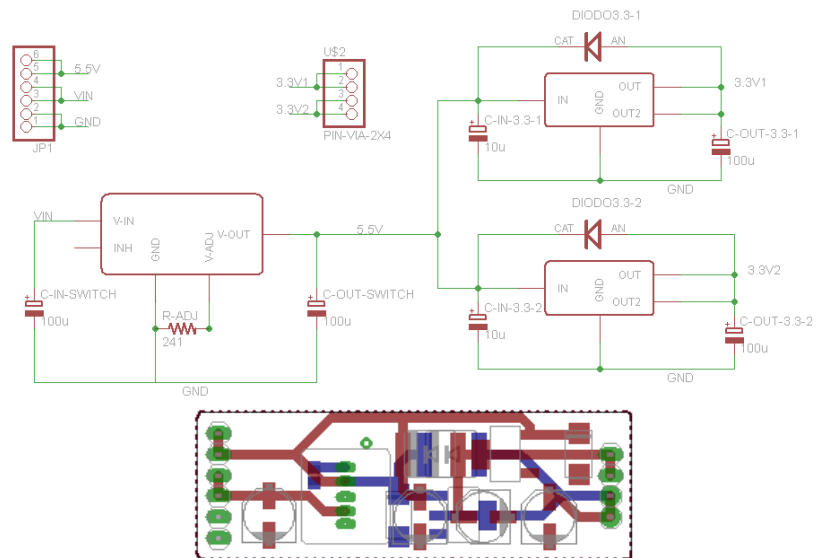


Figura 1: Esquemático y placa módulo alimentación

Módulo conmuta/decod

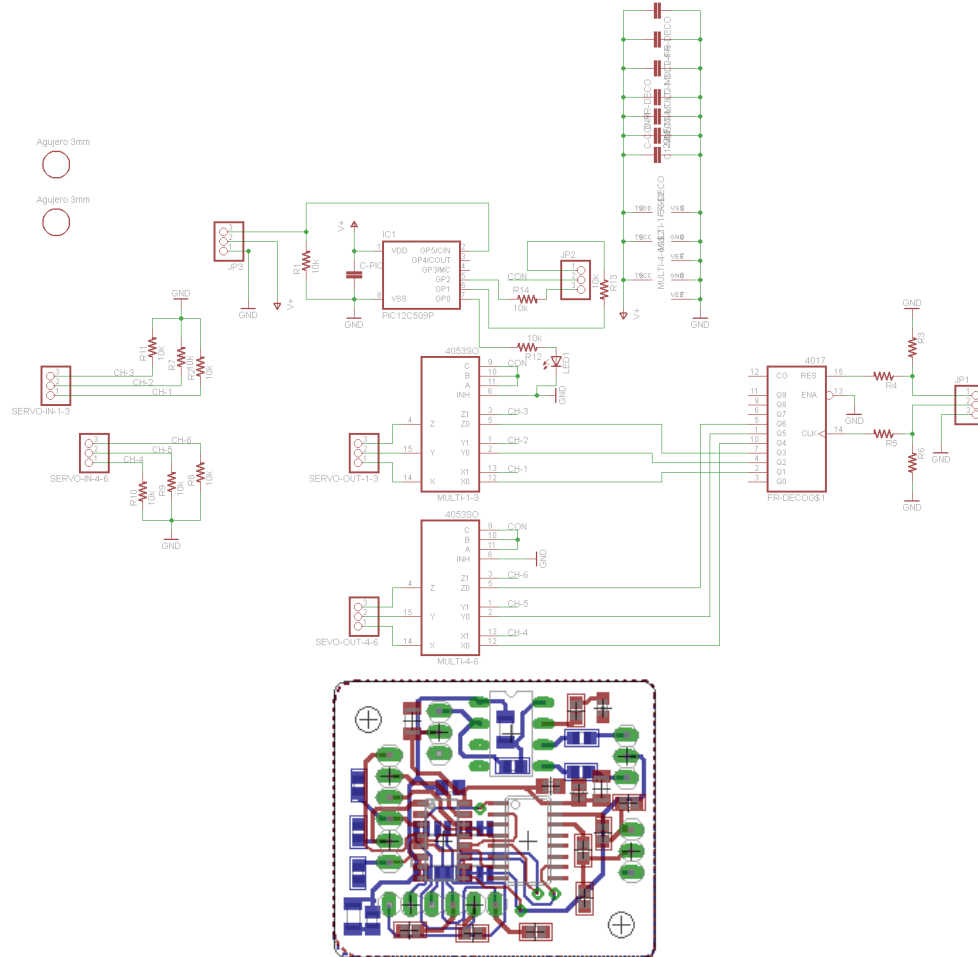


Figura 2: Esquemático y placa módulo conmuta/decod

Módulo IR

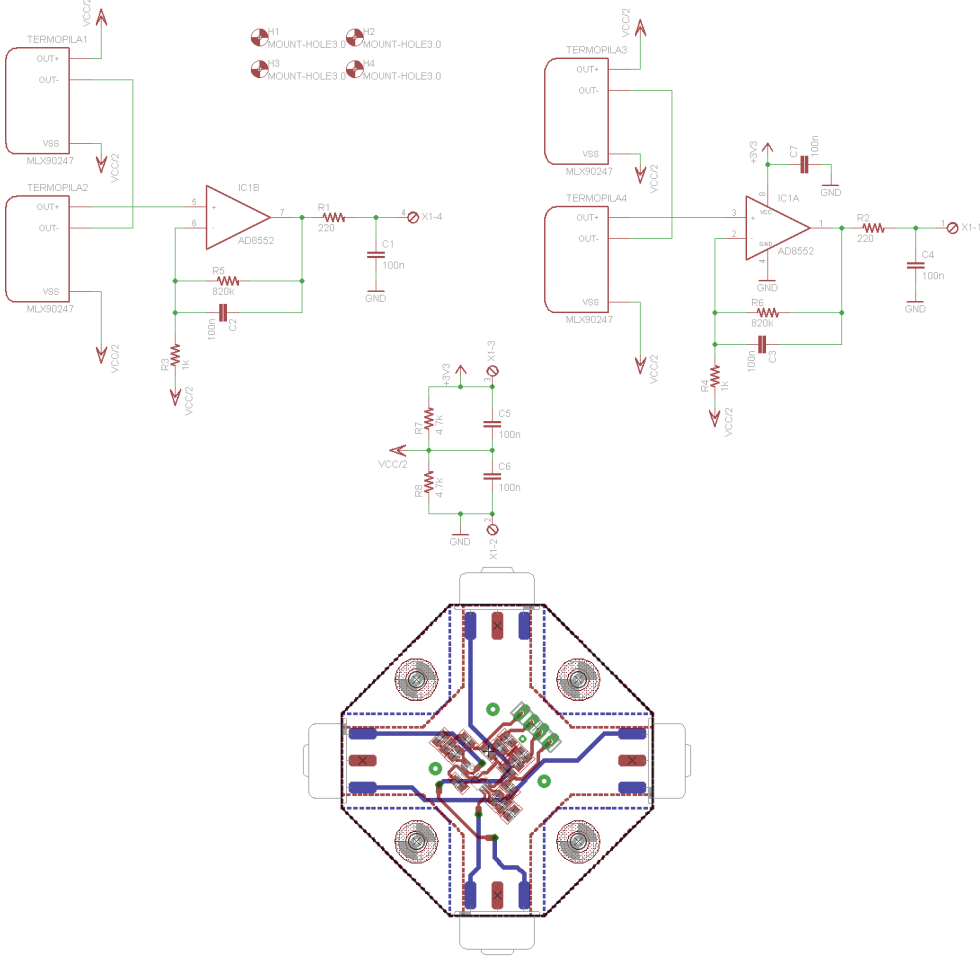


Figura 3: Esquemático y placa módulo IR

Placa madre

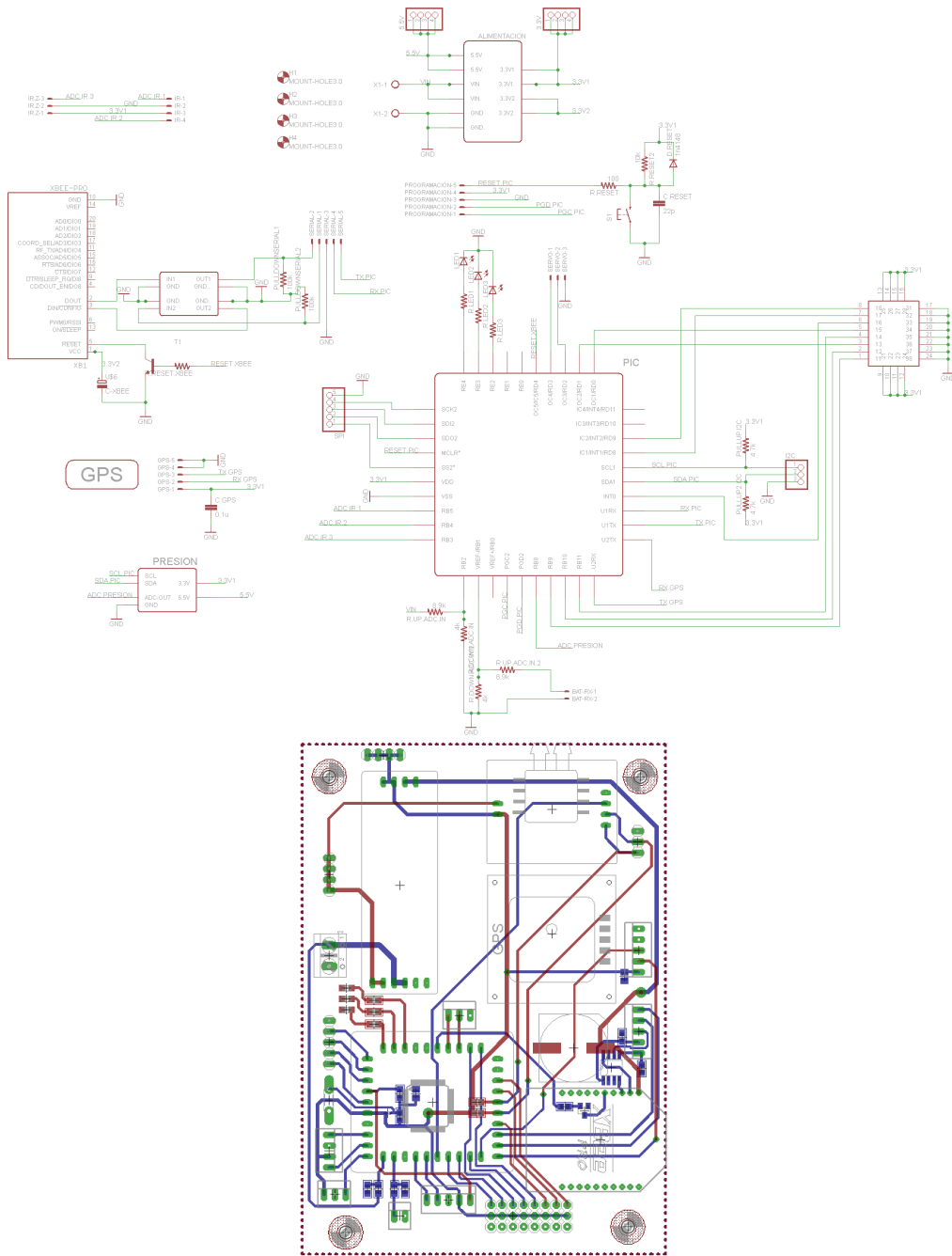


Figura 4: Esquemático y placa madre

Módulo PIC

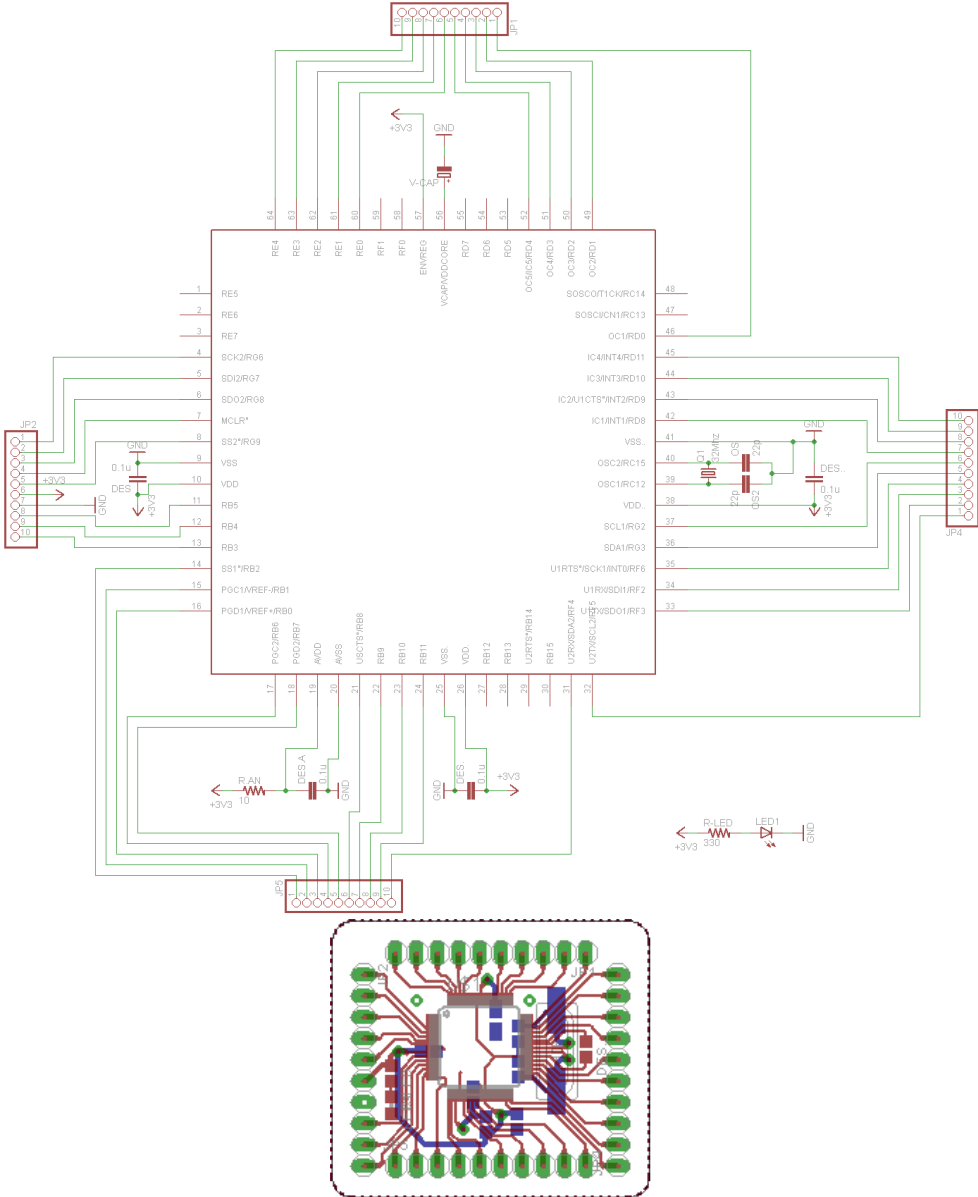


Figura 5: Esquemático y placa módulo PIC

Módulo presión

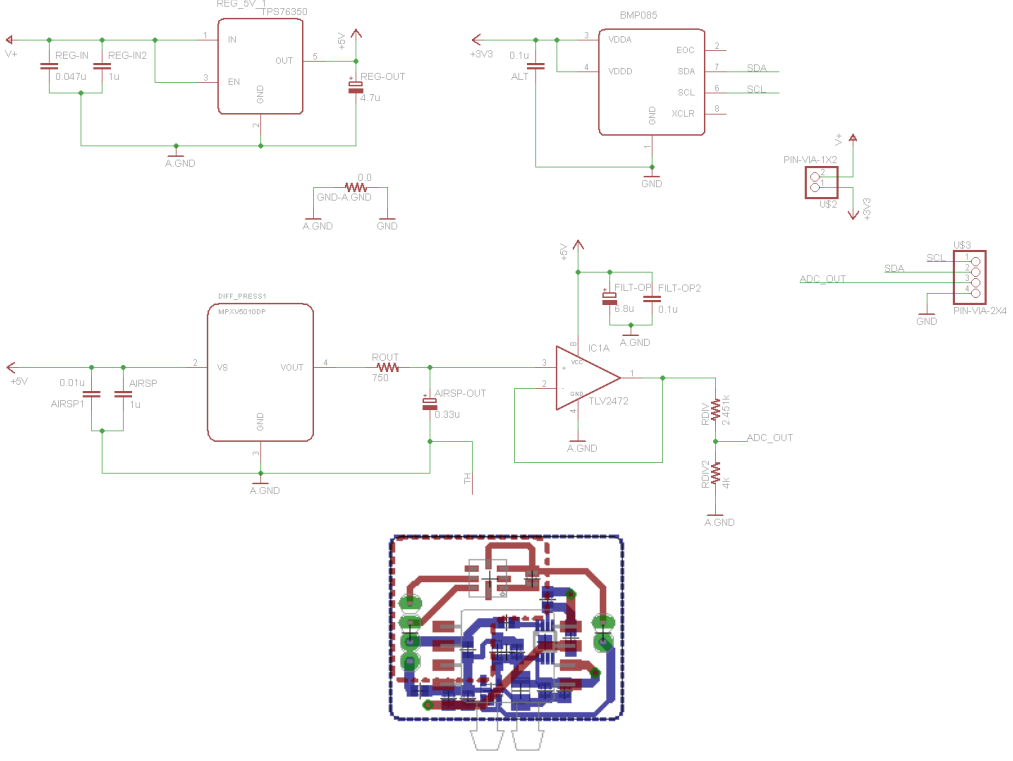


Figura 6: Esquemático y placa módulo presión

Módulo serial-USB

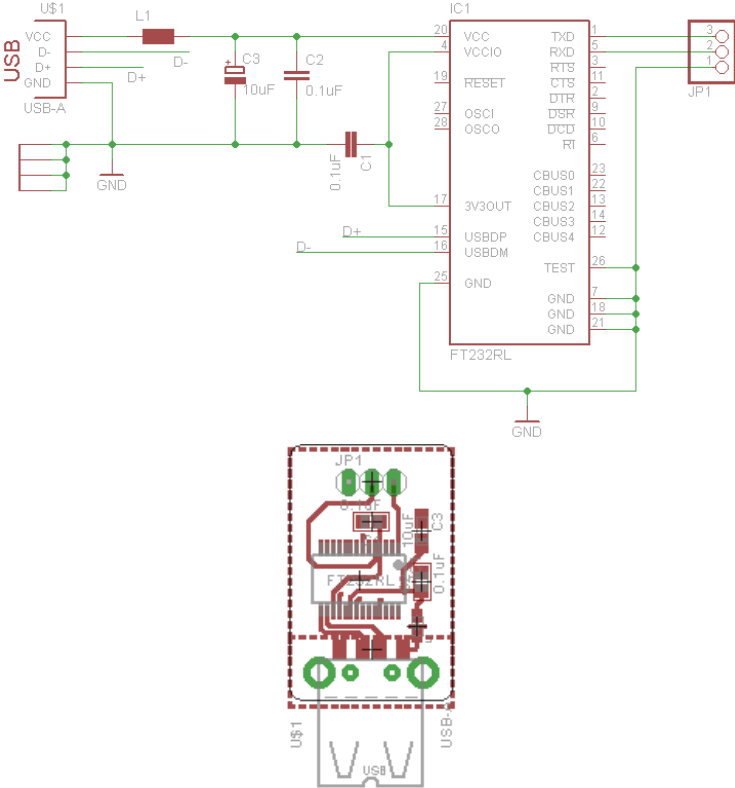


Figura 7: Esquemático y placa módulo serial-USB

Programas realizados

Main toma de datos modo RC

```
#include "configuracion.h"
#include "adcs.h"
#include "gps.h"
#include "i2c.h"
#include "leds.h"
unsigned int16 aux1 = 0;
//!unsigned int8 aux2 = 0;
//!unsigned int8 aux3 = 0;
//!unsigned int8 aux4 = 0;
unsigned int16 contador_ir = 0;
unsigned int16 contador_control = 0;
unsigned int16 contador_gps_presion = 0;
unsigned int16 contador_enviar_ir = 0;
unsigned int16 contador_enviar_gps_presion = 0;
unsigned int16 contador_reset_xbee = 0;
char flag_ir = 0;
char flag_control = 0;
char flag_gps_presion = 0;
char flag_enviar_ir = 0;
char flag_enviar_gps_presion = 0;
char flag_reset_xbee = 0;
char flag_temp_xbee = 0;
unsigned int16 val1;
unsigned int16 val2;
#int_TIMER2
void TIMER2_isr(void)
{
    contador_ir++;
    contador_control++;
    contador_gps_presion++;
    contador_enviar_ir++;
    contador_enviar_gps_presion++;
    contador_reset_xbee++;
}
void main()
{
    setup_adc_ports(sAN1|sAN2|sAN3|sAN4|sAN5|sAN8|VSS_VDD); //seteo del ADC
    setup_adc(ADC_CLOCK_INTERNAL);
    setup_timer2(TMR_INTERNAL|TMR_DIV_BY_64,5000); //interrupcion cada 10ms: (1/16000000)*64*2500
    output_low (rstxbee);
    output_low (servo1);
    delay_ms(1000);
    calibration(); //leo una vez los parametros del bosch para hacer la calibracion
    leds();
    //! fprintf(gps, "$PMTK251,56000*2B\r\n");
    //! delay_ms(500);
    //! fprintf(gps, "$PMTK314,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0*29\r\n"); //seteo solo GPRMC a 1HZ
    //! delay_ms(500);
    //! fprintf(gps, "$PMTK220,1000*1F\r\n");
    //! delay_ms(1000);
    enable_interrupts(INT_RDA2);
    enable_interrupts(INT_TIMER2);
    enable_interrupts(INTR_GLOBAL);
    while(1){
        if(contador_ir == 1){
            contador_ir = 0;
            flag_ir = 1;
        }
    }
}
```

```

if(contador_control == 9){
contador_control = 0;
flag_control = 1;
}
if(contador_gps_presion == 49){
contador_gps_presion = 0;
flag_gps_presion = 1;
}
if(contador_enviar_ir == 3){
contador_enviar_ir = 0;
flag_enviar_ir = 1;
}
if(contador_enviar_gps_presion == 50){
contador_enviar_gps_presion = 0;
flag_enviar_gps_presion = 1;
}
if(contador_reset_xbee == 14000){
contador_reset_xbee = 0;
flag_reset_xbee = 1;
}
if (flag_ir == 1){
val1 = lecturaADC(adcir1);
val2 = lecturaADC(adcir2);
flag_ir = 0;
}
if (flag_enviar_ir == 1){
//! val1 = aux1;
//! val2 = aux1+2;
//! aux1++;
//! if(aux1==2000)
//! aux1=0;
fprintf(serial,"[I|%u|%u]",val1,val2);
flag_enviar_ir = 0;
output_toggle(led1);
}
if (flag_gps_presion == 1){
// fprintf(serial,"%s",gprmcbuffer);
parseo();
fprintf(serial,"[G|%s|%s|%s|%s|",latitud,longitud,velocidad,rumbo);
flag_gps_presion = 0;
}
if (flag_enviar_gps_presion == 1){
cp.upresion = (unsigned int)lecturai2c(presionbosch);
cp.utemperatura = (unsigned int)lecturai2c(tempbosch);
// transformacion();
fprintf(serial,"[P|%Lu|%Lu]",cp.upresion,cp.utemperatura);
flag_enviar_gps_presion = 0;
}
if (flag_reset_xbee == 1){
if (flag_temp_xbee == 0){
output_high(rstxbee);
delay_us(500);
output_low(rstxbee);
flag_reset_xbee = 0;
flag_temp_xbee = 1;
}
else if(flag_temp_xbee == 1){
flag_reset_xbee = 0;
flag_temp_xbee = 0;
}
}
}
}
}
}
}

```

Main toma de datos conmutaciones RC-autónomo

```
#include "configuracion.h"
#include "adcs.h"
#include "gps.h"
#include "i2c.h"
#include "leds.h"
unsigned int16 aux1 = 0;
//!unsigned int8 aux2 = 0;
//!unsigned int8 aux3 = 0;
//!unsigned int8 aux4 = 0;
unsigned int16 contador_ir = 0;
unsigned int16 contador_control = 0;
unsigned int16 contador_gps_presion = 0;
unsigned int16 contador_enviar_ir = 0;
unsigned int16 contador_enviar_gps_presion = 0;
unsigned int16 contador_reset_xbee = 0;
char flag_ir = 0;
char flag_control = 0;
char flag_gps_presion = 0;
char flag_enviar_ir = 0;
char flag_enviar_gps_presion = 0;
char flag_reset_xbee = 0;
char flag_temp_xbee = 0;
unsigned int16 val1;
unsigned int16 val2;
signed int16 alabeo;
signed int16 cabeceo;
float kalab = 2;
float kcabec = 2;
unsigned int16 pulsoservos[9];
char indice = -1;
char bla = 0;
char conmutado = 0;
#int TIMER3
void TIMER3_isr(void)
{
    indice = (indice+1);
    switch (indice)
    {
        case (8):
            set_compare_time(PWM_PIN, 6000, 6000);
            set_compare_time(RESET_PIN, 50, 200);
            *(unsigned int16 *)0x010E = (unsigned int16)(4000);
            indice = -1;
            break;
        case 0:
            set_compare_time(RESET_PIN, 6000, 6001);
            set_compare_time(PWM_PIN, 50, 200);
            default:
            *(unsigned int16 *)0x010E = (unsigned int16)(pulsoservos[indice]);
            break;
    }
}
#int TIMER2
void TIMER2_isr(void)
{
    contador_ir++;
    contador_control++;
    contador_gps_presion++;
    contador_enviar_ir++;
    contador_enviar_gps_presion++;
    contador_reset_xbee++;
}
```

```

}
void control(signed int16 alab,signed int16 cabec);
void main()
{
  setup_adc_ports(sAN1|sAN2|sAN3|sAN4|sAN5|sAN8|VSS_VDD); //seteo del ADC
  setup_adc(ADC_CLOCK_INTERNAL);
  setup_timer2(TMR_INTERNAL |TMR_DIV_BY_64 ,(unsigned int16)5000); //interrupcion cada 20ms:
(1/16000000)*64*5000
  setup_timer3(TMR_INTERNAL |TMR_DIV_BY_64 ,(unsigned int16)5000); //resolución de 0.004ms
  set_compare_time(PWM_PIN, 6000, 6001);
  setup_compare(PWM_PIN, COMPARE_CONT_PULSE | COMPARE_TIMER3);
  set_compare_time(RESET_PIN, 6000, 6001);
  setup_compare(RESET_PIN, COMPARE_CONT_PULSE | COMPARE_TIMER3);
  output_low(rstxbee);
  output_low(pulso);
  output_low(pulsores);
  delay_ms(1000);
  calibration(); //leo una vez los parametros del bosch para hacer la calibracion
  leds();
  delay_ms(500);
  output_high (led2);
  pulsoservos[0] = (unsigned int16)350;
  pulsoservos[1] = (unsigned int16)360;
  pulsoservos[2] = (unsigned int16)370;
  pulsoservos[3] = (unsigned int16)380;
  pulsoservos[4] = (unsigned int16)390;
  pulsoservos[5] = (unsigned int16)400;
  pulsoservos[6] = (unsigned int16)410;
  pulsoservos[7] = (unsigned int16)420;
  pulsoservos[8] = (unsigned int16)190;
  enable_interrupts(INT_TIMER2);
  enable_interrupts(INT_TIMER3);
  enable_interrupts(INTR_GLOBAL);
  while(1){
    if(input(estadoconmuta))
      conmutado = 0;
    else
      conmutado = 1;
    if(contador_ir == 1){
      contador_ir = 0;
      flag_ir = 1;
    }
    if(contador_enviar_ir == 8){
      contador_enviar_ir = 0;
      flag_enviar_ir = 1;
    }
    if(contador_reset_xbee == 14000){
      contador_reset_xbee = 0;
      flag_reset_xbee = 1;
    }
    if (flag_ir == 1){
      val1 = lecturaADC(adcir1);
      val2 = lecturaADC(adcir2);
      cabeceo = ((signed int16)val1 - 525) + ((signed int16)val2 - 525);
      alabeo = ((signed int16)val1 - 525) - ((signed int16)val2 - 525);
      control(cabeceo,alabeo);
      flag_ir = 0;
    }
    if (flag_enviar_ir == 1){
      fprintf(serial, "[I] %u | %d | %d", conmutado, cabeceo, alabeo);
      flag_enviar_ir = 0;
    }
    if (flag_reset_xbee == 1){
      if (flag_temp_xbee == 0){

```

```

output_high(rstxbee);
delay_us(500);
output_low(rstxbee);
flag_reset_xbee = 0;
flag_temp_xbee = 1;
}
else if(flag_temp_xbee == 1){
flag_reset_xbee = 0;
flag_temp_xbee = 0;
}
}
}
}
void control(signed int16 cabec,signed int16 alab)
{
pulsoservos[elevador] = (unsigned int16)((unsigned int16)365 - (signed int16)((float)cabec)*kcabec/4));
if (alab > 0){ //-> inclinado hacia la izquierda !!!
pulsoservos[alonderecho] = (unsigned int16)((unsigned int16)335 + (signed int16)((float)alab)*kalab)/4));
pulsoservos[aleronizquierdo] = (unsigned int16)((unsigned int16)390 + (signed int16)((float)alab)*kalab /
2)/4));
}
else { //-> inclinado a la derecha.....
pulsoservos[alonderecho] = (unsigned int16)((unsigned int16)335 + (signed int16)((float)alab)*kalab /
2)/4));
pulsoservos[aleronizquierdo] = (unsigned int16)((unsigned int16)390 + (signed int16)((float)alab)*kalab/4));
}
}
}
}

```

Configuración

```

#ifndef _CONF_
#define _CONF_
#include <24FJ128GA006.h>
#FUSES NOWDT //No Watch Dog Timer
#FUSES NOJTAG //JTAG disabled
#FUSES NOPROTECT //Code not protected from reading
#FUSES NOWRT //Program memory not write protected
#FUSES NODEBUG //No Debug mode for ICD
#FUSES NOCOE //Device will reset into operational mode
#FUSES ICS1 //ICD communication channel 1
#FUSES NOWINDIS //Watch Dog Timer in Window mode
#FUSES WPRES128 //Watch Dog Timer PreScalar 1:128
#FUSES WPOSTS16 //Watch Dog Timer PostScalar 1:32768
#FUSES IESO //Internal External Switch Over mode enabled
#FUSES PR //Primary Oscillator
#FUSES HS //Oscilador rapido !
#FUSES NOCKSFSM //Clock Switching is disabled, fail Safe clock monitor is disabled
#FUSES NOOSCIO //OSC2 is general purpose output
#device ADC=10
#include <math.h>
#include <stdlib.h>
#use delay(clock=32000000)
#use rs232(UART1,baud=56000,parity=N,bits=8,STREAM=serial)
#use rs232(UART2,baud=56000,parity=N,bits=8,STREAM=gps)
#use I2C(master,I2C1,fast=100000)
#use fixed_io(E_outputs=PIN_E2, PIN_E3, PIN_E4)
#use fixed_io(D_outputs=PIN_D3, PIN_D4)
//Listado de pines:
//RE2 LED3
//RE3 LED2
//RE4 LED1

```



```

//RD2 Servo2
//RD3 Servo1
//RB8 ADC Presion
//RB1 ADC Bat rx
//RB2 ADC Alim
//RB3 ADC IR 3
//RB4 ADC IR 2
//RB5 ADC IR 1
//RD4 Reset XBee
//RB9,RB10,RB11 Header
//RD0,RD1,RD8,RD9 Header
#define led1 PIN_E4
#define led2 PIN_E3
#define led3 PIN_E2
#define servo1 PIN_D3
#define servo2 PIN_D2
#define rstxbee PIN_D4
#define adcalim 2
#define adcrx 1
#define adcir3 3
#define adcir2 4
#define adcir1 5
#define adcpresion 8
#define dirbosch 0xEE
#define presionbosch 0x2E
#define tempbosch 0x34
#define OSS 0
#endif _CONF_

```

Lectura ADC

```

#include "adcs.h"
unsigned int16 lecturaADC(int8 canal)
{
  unsigned int16 lectura;
  set_adc_channel(canal);
  delay_us(20);
  lectura = (int16)(read_adc());
  return lectura;
}

```

Lectura GPS

```

#define SOYGPS
#include "gps.h"
char gprmc[75]; //arreglo que almacena lo que llega del gps, todo el msg GPRMC !!!
char gprmcbuffer[75]; //arreglo buffer que almacena todo cuando esta listo para evitar sobreescrituras
momentaneas
char latitud[9];
char longitud[9];
char velocidad[6];
char rumbo[6];
static char valor = 0;
unsigned int rxgpslisto = 0;
char fix = 0;
char coma[3];
#INT_RDA2
void rda_isr_2() { // Interrupción recepción serie USART
  disable_interrupts(INT_TIMER2);

```

```

valor++;
gprmc[valor] = fgetc(gps);
if (gprmc[valor]==36){ //36 es el ascii de $, detecto cuando empieza una linea.
valor = 0;
// output _ toggle(led1);
}
if (gprmc[valor] == 10){ //busco el final de la linea
// if(rxgpslisto == 0){
// rxgpslisto = 1; //cuando llega el ultimo caracter levanto un flag para avisar que el msg esta completo
memcpy(gprmcbuffer,gprmc,valor); //copio al buffer cuando llega, de aca sacare todo los datos
// fprintf(serial, "%s",gprmcbuffer);
valor = 0;
// }
}
enable_interrupts(INT_TIMER2);
}
void parseo(){ //funcion que parsea el gran string en datos usables....
unsigned int *ptr;
strcpy(coma, ".");
ptr = strtok(gprmcbuffer, coma); //retorna GPRMC
ptr = strtok(0, coma); //hora
ptr = strtok(0, coma); //decimal hora
ptr = strtok(0, coma); //fix status, Aal rumbo
strcat(rumbo,ptr);
rxgpslisto = 0;
}

```

Lectura I2C

```

#define SOYI2C
#include "i2c.h"
struct bosch_BMP805
{
signed int32 ac1;
signed int32 ac2;
signed int32 ac3;
unsigned int32 ac4;
unsigned int32 ac5;
unsigned int32 ac6;
signed int32 b1;
signed int32 b2;
signed int32 mb;
signed int32 mc;
signed int32 md;
unsigned int32 upresion;
unsigned int32 utemperatura;
signed int32 P;
signed int32 T;
} cp;
signed int16 calib_v2(unsigned int8 dirlect)
{
signed int16 dato;
signed int16 msb;
signed int16 lsb;
i2c_start();
delay_ms(100);
i2c_write(0xEE|0x00);
delay_ms(10);
i2c_write(dirlect);
delay_ms(100);
i2c_start();

```

```

delay_ms(10);
i2c_write(0xEE|0x01);
msb = i2c_read(1);
lsb = i2c_read(0);
i2c_stop();
dato = ((msb<<8) | (lsb));
return dato ;
}
void calibration()
{
  /* cp.ac1 = 408; //descomentado funciona mejor
  /* cp.ac2 = -72; //descomentado funciona mejor
  /* cp.ac3 = -14383;
  /* cp.ac4 = 32741; //descomentado funciona mejor
  /* cp.ac5 = 32757;
  /* cp.ac6 = 23153;
  /* cp.b1 = 6190;
  /* cp.b2 = 4;
  /* cp.mb = -32768;
  /* cp.mc = -8711;
  /* cp.md = 2868;
  cp.ac1 = calib_v2(0xAA);
  cp.ac2 = calib_v2(0xAC);
  cp.ac3 = calib_v2(0xAE);
  cp.ac4 = calib_v2(0xB0);
  cp.ac5 = calib_v2(0xB2);
  cp.ac6 = calib_v2(0xB4);
  cp.b1 = calib_v2(0xB6);
  cp.b2 = calib_v2(0xB8);
  cp.mb = calib_v2(0xBA);
  cp.mc = calib_v2(0xBC);
  cp.md = calib_v2(0xBE);
  fputc(252,serial);
  fputc(126,serial);
  fprintf(serial,"%Ld~",cp.ac1);
  fprintf(serial,"%Ld~",cp.ac2);
  fprintf(serial,"%Ld~",cp.ac3);
  fprintf(serial,"%Lu~",cp.ac4);
  fprintf(serial,"%Lu~",cp.ac5);
  fprintf(serial,"%Lu~",cp.ac6);
  fprintf(serial,"%Ld~",cp.b1);
  fprintf(serial,"%Ld~",cp.b2);
  fprintf(serial,"%Ld~",cp.mb);
  fprintf(serial,"%Ld~",cp.mc);
  fprintf(serial,"%Ld~",cp.md);
}
unsigned int16 lecturai2c(unsigned int8 regi)
{
  unsigned int16 dato;
  unsigned int16 msb;
  unsigned int16 lsb;
  i2c_start();
  i2c_write(dirbosch|0x00);
  i2c_write(0xF4);
  i2c_write(regi);
  //delay_ms(6);
  i2c_stop();
  delay_ms(6);
  //delay_ms(10);
  i2c_start();
  i2c_write(0xEE|0x00);
  i2c_write(0xF6|0x00);
  //delay_ms(10);
  i2c_start();

```

```

//delay_ms(10);
i2c_write(0xEE|0x01);
msb = i2c_read(1);
lsb = i2c_read(0);
i2c_stop();
dato = (((msb)<<8) | (lsb));
return dato ;
}
//FUNCIONA LA TRANSFORMACION !
void transformacion()
{
int32 x1,x2,x3,b3,b5,b6,p,t=0;
unsigned int32 b4,b7;
x1=(((cp.temperatura-cp.ac6))*((signed int32)cp.ac5))/0x00008000;
x2=(cp.mc*0x00000800)/(x1+cp.md);
t=(x1+x2+8)/16;
b5 = x1 + x2;
b6 = (signed int32)b5 - 4000;
x1 = (cp.b2 * (b6 * b6 >> 12)) >> 11;
x2 = cp.ac2 * b6 >> 11;
x3 = x1 + x2;
b3 = (((int32) cp.ac1) * 4 + x3) / 4;
x1 = (cp.ac3 * b6) >> 13;
x2 = (cp.b1 * (((signed int32)(b6)) * b6 >> 12)) >> 16;
x3 = ((x1 + x2) + 2) >> 2;
b4 = (cp.ac4 * ((unsigned int32)x3 + (unsigned int32)32768)) >> 15;
b7 = ((unsigned int32)cp.upresion - b3) * (50000);
p = b7 < 0x80000000 ? (b7 * 2) / b4 : (b7 / b4) * 2;
x1 = (p/256)*(p/256);
x1 = (x1 * 3038) >> 16;
x2 = (((signed int32)(-7357)) * p)/(0x10000);
p = p + ((x1 + x2 + 3791)/16);
cp.P = p;
cp.T = t;
}

```