



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

Sistema de Medida del Ángulo de Ataque mediante Sensores de Presión Diferencial

**TRABAJO FINAL DEL
Grado en Ingeniería Aeroespacial**

**REALIZADO POR
Pablo Pérez Correa**

**TUTORIZADO POR
Miguel Alcañiz Fillol
Rafael Masot Peris**

CURSO ACADÉMICO: 2019/2020

Agradecimientos

En primer lugar agradecer a mis tutores Miguel y Rafa su compromiso con este Trabajo. Por el apoyo demostrado desde el primer día, los consejos, las horas dedicadas y su encomiable ayuda continua en el desarrollo del proyecto. *Gracias.*

Quiero agradecer también a mis compañeros del Grado en Ingeniería Aeroespacial de la UPV estos cuatro años compartidos. Muy en especial a Guille, porque no me imagino esta carrera sin su inestimable compañía. *Gracias.*

También a mis amigos, que han estado atentos al desarrollo de este Trabajo, siempre disponibles para ofrecer sus ánimos cuando más se necesitaban y mostrándome su confianza en los momentos de dudas. *Gracias.*

Y, cómo no, a mi familia, en especial a mis padres, por confiar y exigir siempre lo mejor de mí porque, de otra forma, esto no habría sido posible. *Gracias.*

Resumen

El porcentaje de accidentes en aviación general es el mayor entre todas las categorías de aviación. Dándose la mayor parte de estos en las fases de despegue y aterrizaje, debidos a la pérdida del control de las aeronaves, según la FAA (Federal Aviation Administration). Como resultado de los estudios llevados a cabo por la organización estadounidense, se recomendaron medidores de ángulo de ataque como el principal sistema de seguridad para aeronaves de aviación general.

Este Trabajo de Fin de Grado cubre las partes de diseño, fabricación y calibración de un dispositivo capaz de medir el ángulo de ataque de una aeronave. Concretamente, se centra en la parte electrónica del montaje aunque también se tienen en cuenta una serie de principios aerodinámicos que afectan a la medida.

Este dispositivo está formado por dos sensores de presión diferencial, un microcontrolador, que recibe la información de los sensores; y una pantalla que muestra de manera eficiente la información al piloto. Todos estos elementos se encuentran conectados mediante una PCB que mejora la estética del prototipo y facilita su implementación. Además, se añade al sistema una pieza diseñada e impresa en 3D que permite la toma de datos de la forma deseada.

El sistema se pretendía calibrar en el túnel de viento y, debido a la crisis generada a raíz de la Covid-19, no se ha tenido acceso a dichas instalaciones, no pudiendo realizar dichas medidas en las condiciones óptimas. Sin embargo el prototipo lleva a cabo las funciones que de él se esperaban a falta de ser calibrado correctamente.

El objetivo principal de este trabajo es proporcionar al piloto información útil y precisa mediante un sistema de bajo coste y así poder mejorar tanto las operaciones como la seguridad del vuelo en aeronaves ligeras que carezcan de esta clase de sensor.

Resum

El percentatge d'accidents en aviació general és el major entre totes les categories d'aviació. Donant-se la major part d'aquests en les fases d'enlairament i aterratge, debut a la pèrdua del control de les aeronaus, segons la FAA (Federal Aviation Administration). Com a resultat dels estudis duts a terme per l'organització estatunidenca, es recomanaren mesuradors de l'angle d'atac com el principal sistema de seguretat per a les aeronaus d'aviació general.

Aquest Treball de Fi de Grau cobreix les parts de disseny, fabricació i calibratge d'un dispositiu capaç de mesurar l'angle d'atac d'una aeronau. Concretament, se centra en la part electrònica del muntatge tot i que també es té en compte una sèrie de principis aerodinàmics que afecten a la mesura.

Aquest dispositiu està format per dos sensors de pressió diferencial, un microcontrolador, que rep la informació dels sensors; i una pantalla que mostra de manera eficient la informació al pilot. Tots aquests elements es troben connectats mitjançant una PCB que millora l'estètica del prototip i facilita la seua implementació. A més, s'afegeix al sistema una peça dissenyada i impresa en 3D que permet la presa de dades de la manera desitjada.

El sistema es pretenia calibrar en el túnel de vent i, a causa de la crisi generada arran de la Covid-19, no s'ha tingut accés a aquestes instal·lacions, no podent realitzar aquestes mesures en les condicions òptimes. No obstant això el prototip duu a terme les funcions que d'ell s'esperaven a falta de ser calibrat correctament.

L'objectiu principal d'aquest treball és proporcionar al pilot informació útil i precisa mitjançant un sistema de baix cost i així poder millorar tant les operacions com la seguretat del vol en aeronaus lleugeres que manquen d'aquesta classe de sensor.

Abstract

The percentage of accidents in General Aviation is the highest among all the aviation categories. Being the phases of take-off and landing the ones where most of these accidents occur, due to the loss of control of the aircraft, according to the FAA (Federal Aviation Administration). As a result to the studies carried out by the North American organization, the implementation of angle of attack systems were recommended as the main solution to improve the flight safety.

This Final Degree Project covers the design, manufacturing and calibration parts surrounding the development of a system able to measure the aircraft's angle of attack. Specifically, the electronic environment of the system will be deeply described, taking also into account several aerodynamic principles that may affect the correct measure.

The device consists of two differential pressure sensors, a micro controller, that receives the information from the sensors; and one Liquid-Crystal Display that shows the information to the pilot in an efficient way. All these elements are connected thanks to the use of a PCB (Printed Circuit Board) that improves the aesthetics of the system and makes easier its implementation. Furthermore, a 3D-printed probe is designed and added to the prototype, so that the measures can be done in a proper way.

The system was supposed to be calibrated making use of a wind tunnel. However, the pandemic derived from the Covid-19 epidemic, has not allowed the entry to the University, where the experiment should have been carried out. Despite of this, the prototype performs the desired actions correctly, and it is ready for the proper calibration.

The main objective of this project is to offer the pilot really useful and precise information by means of a low-cost system, improving both the in-flight operations and the safety in light aircraft lacking this type of sensor.

Índice general

Agradecimientos	II
Resumen	III
Índice de figuras	IX
Índice de tablas	X
Nomenclatura	XI
1. Introducción	1
1.1. Justificación	2
1.2. Objetivos	6
1.3. Marco Teórico	7
2. Descripción del sistema	11
2.1. Diagrama de bloques	12
2.2. Hardware	14
2.2.1. Arduino UNO	15
2.2.2. Multiplexor PCA9540BD	18
2.2.3. Sensores MS4525DO	20
2.2.4. Pantalla ILI9341	23
2.2.5. PCB	25
2.3. Software	28
2.3.1. IDE Arduino	28
2.3.2. Código	29
2.4. Diseño sonda 3D	35
3. Calibración del sistema	38
3.1. Método de medición	40
3.2. Resultados de la calibración	42

4. Conclusión	45
5. Presupuesto	46
A. Fichas de datos técnicos	48
B. Código calibración	49
Bibliografía	51

Índice de figuras

1.1. Efecto de la altitud en la potencia requerida	2
1.2. Potencia requerida adimensional	2
1.3. Método de la NASA para obtener el ángulo de ataque	4
1.4. Relación entre el ángulo de ataque y las presiones medidas por los sensores, independiente de la presión dinámica	4
1.5. Definición del ángulo de ataque	7
1.6. Sensor real del ángulo de ataque de una aeronave comercial y su localización en el fuselaje	8
1.7. Dispositivo BendixKing KLR10	10
1.8. Dispositivo Garmin GI 260 AoA	10
2.1. Diagrama de bloques del sistema electrónico completo	13
2.2. Placa Arduino UNO Rev3	15
2.3. Pines de alimentación del sistema	16
2.4. Líneas comunicación SPI	17
2.5. Multiplexor I^2C PCA9540BD	18
2.6. Protocolo de comunicación mediante el multiplexor I^2C PCA9540BD	18
2.7. Pinout del multiplexor PCA9540BD	19
2.8. Sensor de presión diferencial MS4525DO	21
2.9. Pinout del sensor de presión diferencial MS4525DO	22
2.10. Esquemático conexión entre el Arduino y los sensores	22
2.11. Pantalla gráfica LCD de 2.8 pulgadas	23
2.12. Esquemático de la conexión entre el Arduino y la pantalla	24
2.13. Interfaz de la pantalla con la información del ángulo de ataque	24
2.14. Esquemático completo del sistema, utilizado para el diseño de la PCB	25
2.15. Detalle del diseño de la PCB con las pistas creadas para las conexiones	26
2.16. PCB diseñada para el montaje de los componentes del sistema	26
2.17. Estructura final del sistema montado sobre la PCB	27

2.18. Dimensión del sistema montado sobre la PCB	27
2.19. Entorno IDE de Arduino	28
2.20. Sketch inicial para el diseño de la pieza 3D	35
2.21. Imagen de la pieza con los surcos creados para los tubos de silicona	36
2.22. CAD final de la pieza 3D diseñada en Fusion 360	37
2.23. Pieza impresa en 3D y sellada con los tubos de silicona en su interior	37
3.1. Sistema completo montado y listo para su calibración	39
3.2. Sonda bajo el ala de una aeronave con dos puertos para medir P_{fwd} y P_{45}	40
3.3. Curva normalizada de la relación entre las presiones y el correspondiente ángulo de ataque, independiente de la presión dinámica	41
3.4. Montaje para la calibración en carretera	42
3.5. Curva de calibración para $V=100$ km/h	43

Índice de tablas

2.1. Rango de presiones para distintas condiciones de vuelo	20
2.2. Resumen de los sensores de presión buscados para implementar en el sistema	20
2.3. Características del MS4525DO-DS-5-A-001-D-P	21
2.4. Pines de conexión entre el Arduino y la pantalla	23
5.1. Costes asociados al material necesario para el sistema	47
5.2. Costes asociados a la mano de obra ingenieril	47

Nomenclatura

Unidades empleadas

g	Gramo
h	Hora
$k\Omega$	Kiloohmio
kB	Kilobyte
km	Kilómetro
mA	Miliamperio
MHz	Megahercio
mm	Milímetro
nF	Nanofaradio
Pa	Pascal
psi	Libra por Pulgada Cuadrada
V	Voltio
$^{\circ}$	Grado sexagesimal
$^{\circ}C$	Grado Centígrado

Magnitudes físicas

α	Ángulo de ataque [$^{\circ}$]
α_{CC}	Ángulo de Carson Cruise [$^{\circ}$]
$\alpha_{EA_{max}}$	Ángulo de máxima eficiencia aerodinámica [$^{\circ}$]
$\alpha_{PR_{min}}$	Ángulo de mínima potencia requerida [$^{\circ}$]
ρ	Densidad del fluido [kg/m^3]
CL	Coefficiente de sustentación [-]
CL_{α}	Pendiente de la curva de sustentación del perfil alar [$(^{\circ})^{-1}$]
L_w	Sustentación producida por las alas [N]
P_{45}	Puerto de presión localizado a 45° de P_{fwd}
P_{Din}	Presión dinámica del fluido [Pa]
P_{Est}	Presión estática del fluido [Pa]

P_{fwd} Puerto de presión perpendicular a la corriente

P_{Tot} Presión total del fluido [Pa]

S_w Superficie alar [m²]

V Velocidad del aire relativo [m/s]

Siglas

AG Aviación General

EASA European Aviation Safety Agency

FAA Federal Aviation Administration

GUI Graphical User Interface

I²C Inter-Integrated Circuit

ICSP In Chip Serial Programmer

IDE Integrated Development Environment

LCD Liquid-Crystal Display

LED Light-Emitting Diode

MISO Master In Slave Out

MOSI Master Out Slave In

NASA National Aeronautics and Space Administration

PCB Printed Circuit Board

SCK/L Signal Clock

SDA Signal Data

SPI Serial Peripheral Interface

SS Slave Select

USB Universal Serial Bus

1. Introducción

El presente Trabajo de Fin de Grado tiene como objetivo el desarrollo de un sistema de bajo coste basado en sensores de presión diferencial capaz de medir el ángulo de ataque de una aeronave.

El Trabajo se estructura de la siguiente forma. En el primer capítulo, llamado *Introducción 1*, se expone la justificación del tema seleccionado para el Trabajo. Este consiste en una descripción del proyecto a desarrollar, de los objetivos del mismo y del marco teórico detrás de los dispositivos destinados a la medida del ángulo de ataque.

En el segundo capítulo, denominado *Desarrollo 2* se explica de forma detallada el diseño del sistema. Se encuentra en él una exhaustiva descripción del hardware y software utilizado. Además, se explica el funcionamiento de cada uno de los componentes, la comunicación entre ellos y la formulación del código basado en C a través del IDE de Arduino para el control de la electrónica del sistema. También se incluye el diseño 3D de la sonda empleada para la toma de datos de las correspondientes presiones.

En el apartado de *Calibración del sistema 3* se explica el método llevado a cabo para la toma correcta de los datos de ambos sensores. Además se muestra la calibración realizada mediante un experimento alternativo al del túnel de viento ante la imposibilidad de acceder a las instalaciones correspondientes. En ella se puede ver el montaje del sistema al completo, similar al que se esperaría en su aplicación final en una aeronave ligera.

Finalmente, en la *Conclusión 4*, se resume el trabajo realizado, desde la fase de diseño hasta la calibración. Se extraen conclusiones sobre las limitaciones del sistema y se expone el trabajo a continuar realizando para implementar el sistema en una aeronave real. Se incluye, como último capítulo el *Presupuesto 5* desglosado del desarrollo del sistema en aplicaciones comerciales.

1.1. Justificación

El ángulo de ataque (α) resulta un parámetro esencial en el vuelo de cualquier aeronave. Este es necesario para la correcta operación de la misma pues es el actor principal en su equilibrado para unas condiciones específicas de vuelo.

El hecho de poder acceder a información precisa sobre α resulta de gran importancia tanto para la seguridad como para la operación efectiva de la aeronave.

En primer lugar, conocer el valor de este parámetro puede ser crucial a la hora de prevenir posibles accidentes, especialmente en las fases de despegue y de aterrizaje. Estas suelen llevarse a cabo a bajas velocidades y altos valores de α , lo que puede facilitar la entrada en pérdida de la aeronave que, en muchos de los casos, resulta imposible de recuperar. Esta importancia queda reflejada, por ejemplo, en las circulares de EASA [1] y de la FAA [2] con el fin de concienciar a los pilotos de cómo reaccionar ante estas situaciones críticas.

Sin embargo, estas no son las únicas etapas relevantes del vuelo. Un ejemplo de esto podría ser la actuación necesaria ante un fallo del motor o una situación de escasez de combustible. En ese caso, el piloto podría tratar de volar la aeronave de forma que el rango de planeo fuera máximo o que el consumo de combustible fuera el mínimo posible. Existen, para ello, algunos ángulos específicos. En primer lugar, el ángulo de máxima eficiencia aerodinámica ($\alpha_{EA_{max}}$), que corresponde con el máximo rango y, en el caso de una aeronave propulsada por un motor de pistón, también con la máxima distancia de planeo. Por otro lado, el ángulo de mínima potencia requerida ($\alpha_{PR_{min}}$), que se corresponde con la máxima autonomía de la aeronave. Otro de los ángulos a conocer es el de *Carson Cruise* (α_{CC}), representando este el ángulo necesario para volar rápido sin conllevar un aumento del consumo de combustible.

Además, estos ángulos son independientes de parámetros como la altitud de vuelo, el peso de la aeronave, la configuración de la misma o el factor de carga, entre otros. Esto se puede ver en las *Figuras 1.1 y 1.2* donde se muestran los gráficos obtenidos en el estudio sobre el "Fundamental angle of attack" [3] y quedan representados los ángulos explicados con anterioridad.

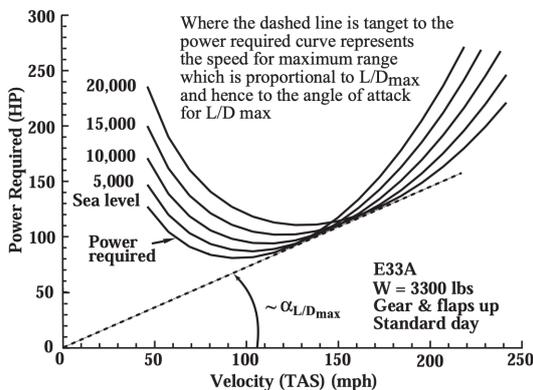


Figura 1.1: Efecto de la altitud en la potencia requerida

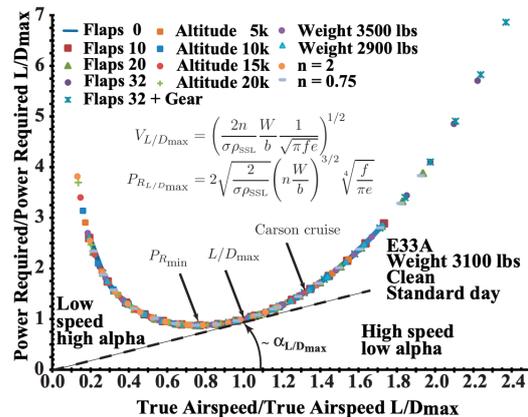


Figura 1.2: Potencia requerida adimensional

Así pues, resulta razonable pensar que el acceso a la información de α en cualquiera de las situaciones expuestas supondría una gran ventaja para poder pilotar la aeronave

correctamente y mejorar la efectividad de sus operaciones.

En la actualidad, las aeronaves ligeras de Aviación General (AG) carecen de sensores que ofrezcan el valor del ángulo de ataque en tiempo real. La gran mayoría de los sistemas que emplean sensores de presión diferencial se centran en las fases de despegue y aterrizaje, es decir, cerca de las condiciones de pérdida. Sin embargo, resultan poco precisos para otras etapas del vuelo que pueden también resultar de interés. Por esto, se les considera más bien dispositivos de alerta de entrada en pérdida (o *stall warning devices* en inglés).

Además, al inconveniente de los dispositivos actuales de servir solo como sistemas de alerta de entrada en pérdida, se suma su alto coste. Dificultando así la adquisición de los mismos por los propietarios particulares de este tipo de aeronaves.

Para comprender mejor el funcionamiento de este tipo de sensores, cabe destacar que la presión total que ejerce un fluido se puede descomponer en su presión estática y su presión dinámica según la *Ecuación 1.1*. Siendo la primera independiente de la velocidad del mismo y la segunda la presión debida a dicha velocidad.

$$P_{Tot} = P_{Est} + P_{Din} \quad (1.1)$$

Siendo la presión dinámica:

$$P_{Din} = \frac{1}{2}\rho V^2 \quad (1.2)$$

Todos los sensores de presión son relativos. Esto se debe a que todos están compuestos por una membrana que siente la presión ejercida que se pretende medir y la compara con otra presión de referencia.

Dependiendo de esta presión de referencia, se pueden distinguir tres tipos de sensores de presión:

- **Sensores de presión absoluta:** Comparan la presión con la del vacío, obteniendo de esta forma la P_{Tot} a la que hace referencia la Ecuación 1.1.
- **Sensores de presión relativa:** Comparan la presión con la presión atmosférica, obteniendo así la diferencia entre ambas; es decir, mide la P_{Din} de la Ecuación 1.1.
- **Sensores de presión diferencial:** Por último, estos sensores están formados por dos entradas, con lo que son capaces de obtener la diferencia de presión entre dos puntos que el usuario requiera para su aplicación.

Si se tiene esto en cuenta, existe una dificultad añadida en la creación de un sistema preciso y fiable capaz de obtener el valor de α . La Federal Aviation Administration (FAA) redactó en diciembre de 2011 una carta [4] con una serie de pautas para la instalación de cualquier sistema referido a este propósito. Entre varias indicaciones que tratan de no alterar la aeronavegabilidad de la aeronave, se encuentra la limitación de que el sistema no debe tener acceso ni al tubo de Pitot ni al indicador de velocidad.

Este hecho supone un cierto inconveniente a la hora del diseño ya que, como se ha visto en la Ecuación 1.2, existe una dependencia entre la presión dinámica que los sensores

diferenciales podrían medir y la velocidad de vuelo. Al no tener acceso a este valor, se deberá de seguir un método alternativo para el desarrollo del sistema.

Como consecuencia, surge la necesidad de encontrar una solución precisa y asequible a esta situación: Un instrumento confiable que sería útil en aeronaves ligeras carentes de este sensor.

Son varios los enfoques que se han planteado con el fin de dar solución al problema. De entre dichas opciones, destaca la desarrollada por la NASA (National Aeronautics and Space Administration) mediante la combinación de sensores de presión en un tubo de Pitot como muestra la *Figura 1.3*.



Figura 1.3: Método de la NASA para obtener el ángulo de ataque

El método consiste en la calibración del sistema con el objetivo de obtener una función que calcule el ángulo de ataque a partir de las diferencias de presión medidas entre P_1 y P_2 . Sin embargo, este no permite evitar la dependencia con la velocidad de vuelo, por lo que, al no tener acceso al medidor de velocidad de la aeronave, no se soluciona el problema.

Este Trabajo de Fin de Grado pretende exponer una solución rigurosa y asequible a esta problemática: el desarrollo de un sistema capaz de calcular el ángulo de ataque de manera precisa mediante dos sensores de presión diferencial de coste económico.

Para ello, se propone el uso de dos sensores MS4525DO que serán montados según el estudio de David F. Rogers [5] como se explicará posteriormente en profundidad. Con ello se espera obtener una curva que relacione la diferencia de presión entre los sensores y el ángulo de ataque correspondiente que sea independiente de la presión dinámica, tras una correcta calibración en el túnel de viento. La *Figura 1.4* muestra el resultado obtenido de aplicar dicho método en el artículo de la FAA de la referencia [6].

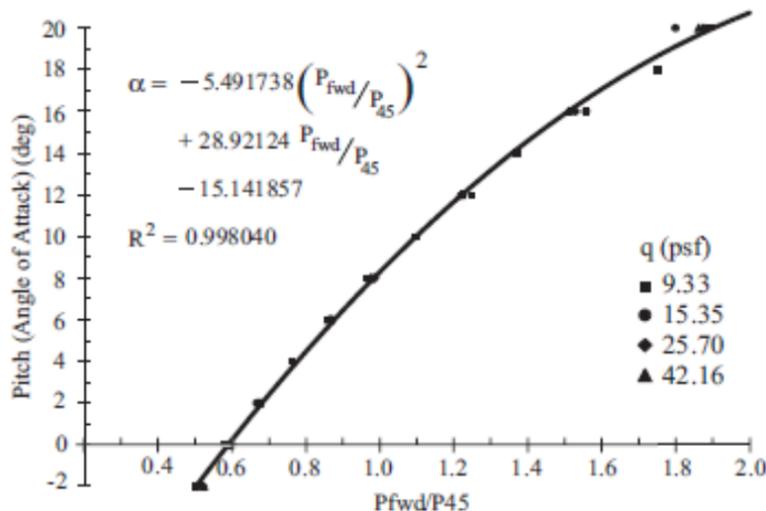


Figura 1.4: Relación entre el ángulo de ataque y las presiones medidas por los sensores, independiente de la presión dinámica

Esta curva servirá también como aproximación para el cálculo de la precisión requerida por los sensores para ofrecer la medida precisa deseada.

Por otra parte, se programa, en una pantalla LCD gráfica basada en el procesador ILI9341, una interfaz de manera que la información medida por el sistema se transfiera de manera efectiva al piloto en cabina.

El desarrollo de todo el sistema electrónico es posible gracias a la conocida placa de desarrollo Arduino UNO. Esta se encarga de coordinar y procesar los datos obtenidos a través de los sensores sobre el valor del ángulo de ataque mediante una función característica conseguida tras la calibración llevada a cabo.

Por último, se emplea el multiplexor I^2C PCA9540BD que permite seleccionar el sensor con el que desea comunicarse el microcontrolador para recibir la información.

Cabe destacar de nuevo que el objetivo principal es la creación del entorno electrónico así como el montaje de un prototipo capaz de realizar las medidas mencionadas del ángulo de ataque y realizarlo con el mayor ratio precisión-precio posible.

1.2. Objetivos

El presente Trabajo Final de Grado tiene como objetivo principal la producción del sistema electrónico necesario para desarrollar un método capaz de medir y mostrar por pantalla el valor del ángulo de ataque de una aeronave mediante el uso de dos sensores de presión diferencial.

Es, por tanto, tarea de este Trabajo proporcionar mediciones de este parámetro de manera eficiente y precisa a través de dichos sensores que se explicarán en mayor profundidad en la sección 2 *Descripción del sistema*. De esta forma, se puede ofrecer información útil al piloto que sirva como referencia a la hora de realizar las actuaciones pertinentes en vuelo. Durante el desarrollo del Trabajo varios son los objetivos propuestos:

- Analizar los dispositivos actuales empleados en funciones relacionadas con la medida del ángulo de ataque.
- Proponer ideas a desarrollar para la completa implementación en una aeronave real. Teniendo en cuenta la viabilidad del montaje para desempeñar su función de medida de manera precisa y cumpliendo con las indicaciones de seguridad estipuladas por las autoridades.
- Desarrollar el sistema de medida completo e independiente, de manera que pueda ser calibrado y posteriormente implementado en una aeronave ligera.
- Describir de forma detallada el hardware y software que conforman el sistema diseñado para conseguir el correcto funcionamiento del mismo.
- Analizar los diferentes sensores de presión diferencial utilizados en la actualidad y realizar un estudio comparativo entre ellos. Se mostrarán sus ventajas e inconvenientes y se justificará la selección de los sensores MS4525DO para la realización de este Trabajo.
- Calibrar correctamente los dos sensores en el túnel de viento con el fin de mejorar la fiabilidad del sistema en la medida y alcanzar la precisión deseada mediante un buen ajuste a la curva de calibración obtenida.

1.3. Marco Teórico

Como se ha mencionado con anterioridad, el objetivo principal de este Trabajo es desarrollar un sistema de ayuda capaz de medir el ángulo de ataque en vuelo de una aeronave mediante el uso de sensores de presión diferencial. Se trata de un planteamiento del cual se esperan resultados precisos mediante una tecnología de coste económico que pueda servir de manera útil al piloto en tiempo real.

Para entender el enfoque del Trabajo es necesario identificar primero qué teorías y modelos fundamentales existen en relación a la medida de α en aeronaves.

Como se ha explicado en la parte de *Introducción 1*, el ángulo de ataque es uno de los parámetros básicos a tener en cuenta en el vuelo de una aeronave. Este representa el ángulo que existe entre la cuerda del perfil alar y la dirección del viento relativo a la misma (ver *Figura 1.5*).

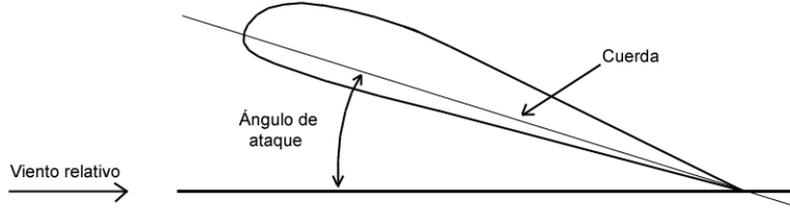


Figura 1.5: Definición del ángulo de ataque

Además está relacionado directamente con la sustentación generada por las alas (L_w) mediante la *Ecuación 1.3*, por lo que permite mantener vuelo equilibrado (sustentación igual al peso de la aeronave) para unas condiciones requeridas por el piloto.

$$L_w = \frac{1}{2} \rho S_w V^2 CL \quad (1.3)$$

Donde ρ es la densidad del aire a la altitud de vuelo, S_w es la superficie alar, V es la velocidad de vuelo de la aeronave y CL es el coeficiente de sustentación, que para el caso de las alas se corresponde con:

$$CL = \alpha CL_\alpha \quad (1.4)$$

Siendo CL_α la pendiente de la curva de sustentación del perfil alar y α el ya mencionado ángulo de ataque. Como se puede ver, para unas mismas condiciones de altitud, superficie alar y velocidad de vuelo, se requerirá un mayor valor de α para equilibrar un peso mayor.

Con el fin de acceder a la información sobre este parámetro durante el vuelo, las aeronaves comerciales cuentan con sensores destinados específicamente a su medida. Este es el caso del ejemplo de la *Figura 1.6*.



Figura 1.6: Sensor real del ángulo de ataque de una aeronave comercial y su localización en el fuselaje

Como se puede ver, se trata de una aleta con forma de perfil alar, normalmente localizada en un lateral del fuselaje de la aeronave donde el flujo se encuentra menos perturbado. Esta se alinea con el viento relativo y transmite, en función de la rotación experimentada, una señal eléctrica al indicador de la cabina que la traducirá y mostrará en pantalla el valor correspondiente de α . Este sensor también dispone de un dispositivo de amortiguación que permite obtener una respuesta dinámica satisfactoria.

Además de la información sobre α , los sensores como el de la *Figura 1.6* sirven también como dispositivos de alerta de entrada en pérdida. Para ello, la información del ángulo de ataque se procesa en el ordenador de abordo junto con la obtenida por el indicador de velocidad de la aeronave y, en el caso de pérdida, se emite un zumbido o una luz en el panel de control que alerta al piloto de la situación.

Como se comentó en la sección de *Justificación 1.1*, gran parte de las aeronaves ligeras que conforman el grupo de Aviación General carecen de este tipo de sensores que ofrezcan información detallada del ángulo de ataque. Sin embargo, sí que poseen el sistema de aviso de entrada en pérdida. En este sentido, varias son las tecnologías implementadas para su funcionamiento.

Una de ellas está compuesta por una pequeña lengüeta que sobresale del borde de ataque. Para ángulos de ataque normales, la lengüeta se coloca de modo que se encuentra hacia abajo dejando la ranura tras ella destapada. A medida que aumenta el ángulo de ataque, las líneas de corriente actúan para levantar la lengüeta y cerrar la ranura, activando así el aviso en cabina de entrada en pérdida, ya sea mediante una señal sonora o mediante la vibración de los controles del piloto.

Otras, por su parte, se basan en el principio aerodinámico del "*buffeting*", o bataneo en castellano, por el que el empenaje trasero vibra de manera notoria justo antes de que entre en pérdida el ala. Esto se debe a que, en estas condiciones, el flujo sobre el extradós del ala no se encuentra adherido a la misma, creándose un flujo turbulento que produce la vibración de la cola, advirtiendo así al piloto.

Como ocurre con cualquier sistema del entorno de una aeronave, estos sensores están sujetos a una cierta regulación. Pese a no ser el proceso de certificación uno de los objetivos del presente Trabajo, se recoge a continuación parte de la regulación relacionada con el sistema de aviso de entrada en pérdida en aeronaves ligeras según la parte CS-23 del

Reglamento de EASA [7].

- Debe existir un sistema de alerta de entrada en pérdida tanto para vuelo recto como en giros y para cualquier configuración de la aeronave (limpia, con flaps y/o tren de aterrizaje desplegados, etc.).
- Las indicaciones emitidas deben ser claras y sin ambigüedades, no debiendo necesitar la alerta de la atención constante del piloto.
- El aviso de entrada en pérdida debe iniciarse como tarde cuando la velocidad sea 10 km/h superior a la de entrada en pérdida, con el fin de que el piloto tenga tiempo de reacción.

Por su parte, la FAA redactó en 2014 un memorándum [8] en el que recogían los requisitos para la instalación de un sensor del ángulo de ataque. En este, se especifica que el uso de dicho sistema debe ser exclusivamente como suplemento de información. Entre las indicaciones, destacan las siguientes:

- La indicación errónea o fallo del sistema no conlleva efectos negativos en la seguridad del vuelo o del piloto.
- Se debe realizar un test que compruebe que el funcionamiento normal del sensor no entra en conflicto con el sistema certificado de aviso de entrada en pérdida de la aeronave. Esto es, deberá ser conservativo en comparación con este.
- Como ya se estipuló previamente según la carta de la FAA mencionada [4], el sistema no debe interferir en ninguno de los instrumentos de medida certificados e instalados en la aeronave (tubo de Pitot, indicador de velocidad, etc.) a excepción de la fuente que lo alimente eléctricamente.

A continuación se muestran dos ejemplos de los sistemas más utilizados para la ayuda en la medida del ángulo de ataque que cuentan con la certificación necesaria para su implementación.

BendixKing KLR 10

En primer lugar se muestra el BendixKing KLR 10. Se trata de un sistema que ofrece de manera visual información sobre la diferencia entre el estado actual de la aeronave y la posible entrada en pérdida de la misma.

Está compuesto por una sonda de presión de aire que se monta en una placa de inspección situada en la parte inferior del ala. Este se conecta a un dispositivo que se adhiere al parabrisas dentro de la cabina (ver *Figura 1.7*), permitiendo al piloto visualizarlo durante el vuelo.



Figura 1.7: Dispositivo BendixKing KLR10

Como se puede ver, posee varias líneas horizontales que informan al piloto de lo lejos que está de la entrada en pérdida. Cuando el avión está cerca de esa situación, la flecha roja se ilumina indicando al piloto que baje el morro del avión, al mismo tiempo que emite alertas sonoras. La principal posible desventaja de este tipo de medidor es que suele ser preciso exclusivamente para una configuración dada de la aeronave, normalmente en la fase de aproximación. Su precio ronda los 1600 \$ [9].

Garmin GI 260 AoA

Por último, se describe el sistema de medida Garmin GI 260 AoA [10]. Al igual que el modelo anterior, este se compone de una sonda de presión que se localiza en la parte exterior de la aeronave y de una pantalla a instalar en el parabrisas de la cabina, cuya interpretación es, también, análoga a la del KLR 10, como se puede ver en la *Figura 1.8*.



Figura 1.8: Dispositivo Garmin GI 260 AoA

A diferencia del anterior, este sistema está normalizado, pues añade a la medida de la diferencia de presión un puerto para obtener la presión estática, con la que calcula la velocidad y la densidad del aire. De esta manera, la medida es mucho más precisa e independiente de la configuración de la aeronave. Su precio varía entre 1500 y 1900 \$.

Como se puede ver, ambos tienen un alto precio, por lo que este Trabajo intentará crear un prototipo más económico a la vez que fiable.

2. Descripción del sistema

En este segundo capítulo se explica en profundidad el sistema diseñado. Con este objetivo se ha dividido en cuatro secciones diferenciadas: diagrama de bloques, hardware, software y diseño de la sonda 3D.

En primer lugar se muestra el diagrama del sistema donde figuran los componentes utilizados y se representan las conexiones de entrada y salida que permiten su correcta comunicación.

Tras ello, en la sección *2.2 Hardware*, se realiza una exhaustiva descripción del hardware utilizado. Concretamente, de la placa Arduino UNO, los sensores de presión diferencial MS4525DO, el multiplexor I^2C PCA9540BD y la pantalla gráfica LCD basada en el procesador ILI9341. También se muestra el diseño de la PCB (Printed Circuit Board) que facilita la conexión entre los componentes anteriores. De ellos, se explicará su principio de funcionamiento, así como las características por las que se han escogido, haciendo especial énfasis en sus beneficios y limitaciones. Toda esta información será extraída de sus correspondientes fichas técnicas (*Datasheet*) que se pueden encontrar en el *Anexo A*.

En el apartado *2.3 Software*, se expone el entorno de desarrollo integrado de Arduino (IDE). Este será el encargado de realizar las funciones del sistema mediante la obtención de valores, comunicación entre elementos y procesamiento de los datos obtenidos. Se presenta en esta apartado, también, el código programado para llevar a cabo dichas funciones, estando este comentado para su fácil comprensión.

Finalmente, en la sección *2.4 Diseño de la sonda*, se muestra el proceso de diseño de la pieza 3D encargada de permitir la correcta toma de los valores de presión requeridos por el sistema. Ambos elementos, el sistema y la sonda, se conectarán mediante dos tubos de silicona, dando por concluido el diseño del sistema.

2.1. Diagrama de bloques

El diagrama de bloques de la *Figura 2.1* pretende representar gráficamente el funcionamiento del sistema desarrollado. Para ello, se muestran en él todos los componentes que lo conforman y el tipo de comunicación entre ellos, así como las conexiones entre los pines de entrada y salida.

- **Placa Arduino UNO Rev3:** es el componente principal del sistema. Contiene el microcontrolador y la memoria flash donde se guarda y ejecuta el programa, así como todos los pines de entrada / salida donde se conecta el resto de elementos que conforman el sistema. Ejerce de maestro de la comunicación tanto con la pantalla como con los sensores: da las órdenes y estos dispositivos realizan las acciones requeridas. Además, sirve como fuente de alimentación al resto de componentes que se conectan al pin de 5 V de salida de este.
- **Sensores de presión diferencial MS4525DO:** encargados de obtener la medida de la presión dinámica del aire. Esta información es leída por el Arduino a través del multiplexor mediante comunicación I^2C . Están conectados a los dos canales de los que dispone el multiplexor.
- **Multiplexor PCA9540BD:** su función es la de seleccionar el sensor con el que se comunica el Arduino UNO. Se conecta a él mediante los pines A4 y A5 del maestro, destinados a la comunicación I^2C .
- **Pantalla gráfica LCD:** la pantalla utilizada está basada en el controlador ILI9341 que permite mostrar texto y gráficos a color. Esta muestra el valor de la medida al usuario de manera clara y eficaz, junto a una pequeña imagen que ayuda al piloto a su visualización. Se conecta al Arduino mediante el bus SPI para lo que utiliza los pines 9, 10, 11 y 13 del maestro.
- **Batería de 1000mAh:** fuente de energía externa conectada a la placa mediante el puerto serie USB-B a 5V.
- **PCB:** se añade una placa de circuito impreso a la que se sueldan todos los componentes mencionados con el objetivo de estructurar el montaje físico del sistema. Así pues, se conectan a ella la pantalla por una de las caras y el Arduino y los sensores por la otra. De esta forma el conjunto de componentes se sitúa de manera compacta y manejable, ahorrando espacio y permitiendo emplazarlo de manera sencilla en la cabina de la aeronave.

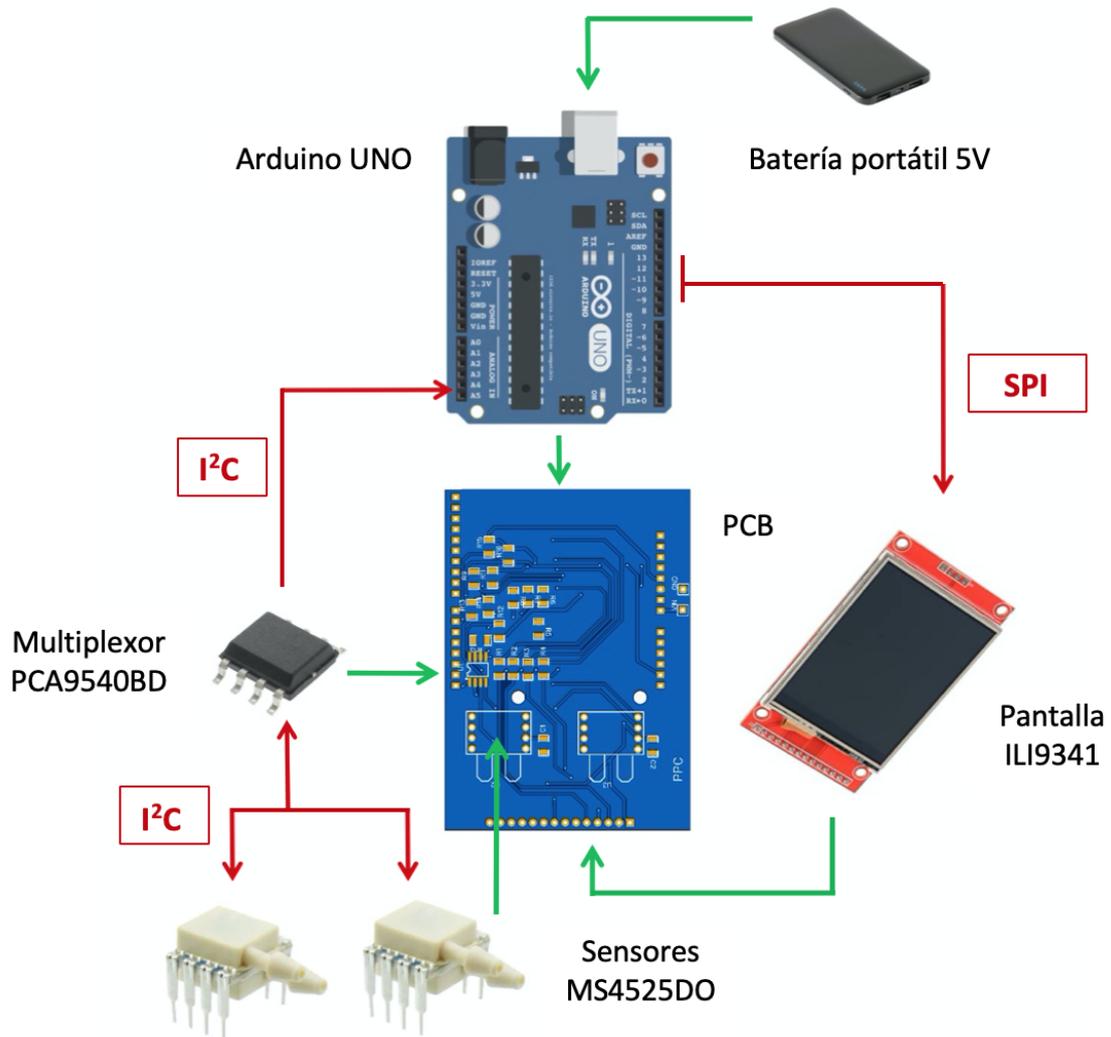


Figura 2.1: Diagrama de bloques del sistema electrónico completo

2.2. Hardware

Como se ha indicado al comienzo de este capítulo *2 Descripción del Sistema*, en esta segunda sección se explica el hardware empleado. Para ello, se detallan a continuación las características de los siguientes componentes: placa Arduino UNO, sensores de presión diferencial, multiplexor I^2C y la pantalla gráfica. Por último, se incluirá una breve descripción de la placa de circuito impreso (PCB) diseñada para su ensamblaje electrónico.

En primer lugar se encuentra la descripción de la placa Arduino UNO Rev3. Esta actúa en el sistema como maestro y es responsable de requerir información de los sensores, procesarla y transmitirla a la pantalla. Por su importancia, se detallan de ella los elementos que la conforman haciendo especial énfasis en sus entradas y salidas y en los protocolos de comunicación empleados con el resto de dispositivos.

A continuación, se exponen los sensores de presión seleccionados, justificando esta decisión. Para ello, se muestra un estudio de la precisión requerida que se espera para la medida y se realiza una comparación con otros sensores disponibles en el mercado.

Con el objetivo de poder acceder adecuadamente a ambos sensores, se añade al sistema el multiplexor I^2C PCA9540BD. Este es el encargado de seleccionar con qué sensor se comunica el Arduino para leer la información obtenida. Se explica de él el protocolo en el que se basa su funcionamiento, extrayendo la información (igual que para el resto de componentes) de su correspondiente ficha técnica.

Para acabar con la parte del hardware empleado se describe la pantalla seleccionada para la función de mostrar la medida realizada. De ella se explica el procesador ILI9341 que permite su programación y se expone también la interfaz que mostrará el valor del ángulo de ataque.

Finalmente se muestran la PCB que permite el montaje del sistema de manera organizada y facilita la implementación del resto de componentes, reduciendo el tamaño del prototipo.

2.2.1. Arduino UNO

Arduino es una plataforma de electrónica de código abierto basado en hardware y software fácil de utilizar. Esto permite una gran flexibilidad para la realización de proyectos por parte de los usuarios, por lo que ha reunido a una gran cantidad de estudiantes, profesores y profesionales de la programación entorno a la plataforma, creando así un sistema conocido en todo el mundo.

Así pues, Arduino ofrece un sistema sencillo y eficaz que aúna las necesidades de hardware y software. Por su parte, el hardware añade al microcontrolador integrado una serie de elementos que permiten la correcta alimentación para su funcionamiento. Estos elementos de alimentación, reguladores, y pines están conectados entre ellos mediante una PCB que conforma la estructura de la placa. En lo referente al software, el entorno de desarrollo de Arduino (IDE) se basa en el lenguaje C. Además, la gran cantidad de librerías generadas y compartidas a la comunidad por los usuarios facilitan enormemente su uso.

Entre la variedad de modelos de placas de distintos tamaños y características desarrolladas, se ha seleccionado para este Trabajo el Arduino UNO Rev3 (ver *Figura 2.2*).



Figura 2.2: Placa Arduino UNO Rev3

Se trata de una placa básica y económica basada en el microcontrolador de 8 bits ATmega328P. Se compone de 14 pines de entrada/salida digitales (parte superior de la *Figura 2.2*) y 6 analógicos (parte inferior derecha de la placa). Contiene un oscilador de cristal de cuarzo que lo hace trabajar a 16 MHz y ofrece una memoria flash de 32KB. Cuenta también con puerto USB que permite su conexión al ordenador o a una batería externa, *power jack*, entrada ICSP, etc. Por otro lado, contiene puertos de comunicación en serie, tanto I^2C como SPI, un LED integrado (conectado al pin 13) y un botón de reset. La placa trabaja a 5V.

En cuanto a la alimentación de la placa, para la exposición de este Trabajo se utiliza la conexión mediante el puerto USB tipo B. Sin embargo, para la implementación en la aeronave, es posible conectarla directamente mediante el *power jack* que regula la tensión de entrada de 7-12V a 5V. Además cabe destacar que el Arduino UNO es el encargado en el sistema diseñado de alimentar al resto de componentes como se ha visto en la sección *2.1 Diagrama de componentes*. Tanto las entradas mencionadas como el pin del que se tomarán los 5V para el resto de componentes quedan representados en la *Figura 2.3*.

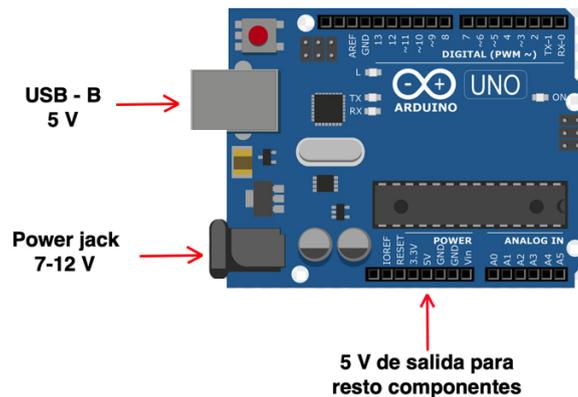


Figura 2.3: Pines de alimentación del sistema

Para finalizar con este subapartado, se explican a continuación los dos protocolos de comunicación en serie utilizados por el sistema: SPI (Serial Peripheral Interface) e I^2C (Inter-Integrated Circuit).

La comunicación mediante el protocolo SPI consta de una estructura del tipo maestro-esclavo. El dispositivo maestro puede iniciar la comunicación con uno o varios dispositivos esclavo, y enviar o recibir datos de ellos. Sin embargo, los esclavos no pueden realizar ninguna de esas dos operaciones. Al realizarse la comunicación entre maestro y esclavos por dos líneas independientes, el maestro puede recibir y enviar información de manera simultánea.

Otra característica del SPI es que es un bus síncrono. Esto disminuye la complejidad del protocolo, ya que el maestro genera una señal de reloj que mantiene a todos los dispositivos sincronizados.

La comunicación SPI requiere de las siguientes líneas:

- **SCK** (Clock Signal): señal de reloj generada por el maestro para sincronizar los dispositivos.
- **MOSI** (Master Out, Slave In): el maestro envía información al esclavo.
- **MISO** (Master In, Slave Out): el maestro recibe información del esclavo.
- **SS** (Slave Select): selecciona el esclavo con el que se realiza la comunicación. Se necesita una línea de selección para cada esclavo conectado con el maestro.

La representación de estas líneas se pueden ver en la *Figura 2.4*:

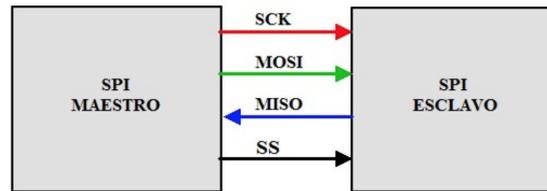


Figura 2.4: Líneas comunicación SPI

En el caso particular de este Trabajo, la comunicación por SPI tendrá lugar entre el Arduino y la pantalla, por lo que solo se requerirá de una única línea SS. La conexión entre estos dos dispositivos se puede observar en el esquemático de la *Figura 2.12*, en la sección 2.2.4.

Por otra parte, a diferencia del bus SPI, la comunicación I^2C requiere únicamente de dos cables para su funcionamiento. Uno para la señal de reloj (SCL) y otro para el envío de datos (SDA). Esto supone una ventaja respecto del bus SPI, si bien es cierto que la electrónica para implementarlo es más compleja.

En el bus I^2C , cada uno de los dispositivos conectados tiene una dirección asignada. El maestro envía la dirección con la que se quiere comunicar para iniciar el protocolo de envío o recepción de datos. El problema principal es que esta dirección de acceso, en la mayoría de ocasiones, viene preestablecida por el fabricante. Por lo tanto cuando se quieren conectar dispositivos del mismo tipo, todos presentan la misma dirección.

En este Trabajo, el bus I^2C se emplea para llevar a cabo la conexión entre la placa Arduino UNO y los dos sensores de presión diferencial. Sin embargo, ambos sensores disponen de la misma dirección de acceso. Con el fin de solucionar este problema de manera eficaz, se opta por implementar en el sistema un multiplexor I^2C que es el objeto del siguiente subapartado.

2.2.2. Multiplexor PCA9540BD

Como se ha comentado en la sección anterior, se opta por añadir al sistema un multiplexor I^2C , concretamente el modelo PCA9540BD (ver *Figura 2.5*). Este se conecta a la placa Arduino UNO y es capaz de seleccionar con cuál de los dos sensores de presión diferencial se va a realizar la comunicación.

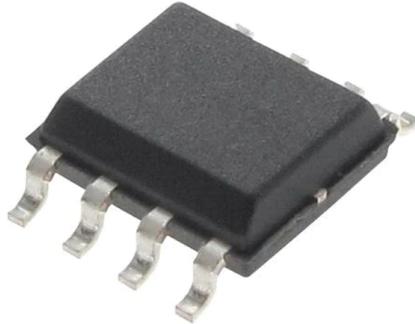


Figura 2.5: Multiplexor I^2C PCA9540BD

Para explicar con claridad el protocolo de funcionamiento del dispositivo se hace uso de la *Figura 2.6*:

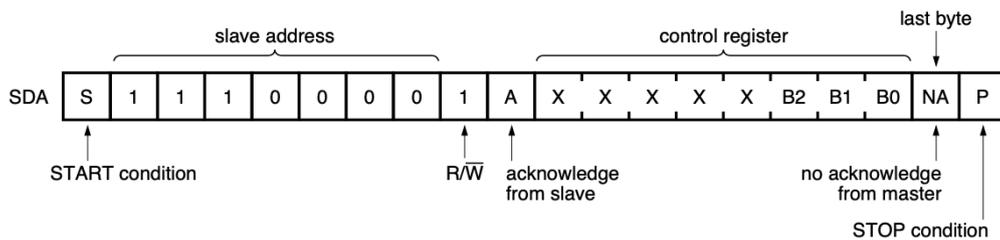


Figura 2.6: Protocolo de comunicación mediante el multiplexor I^2C PCA9540BD

En ella se distinguen varios grupos de bits que el maestro (placa Arduino UNO) debe transmitir por el bus I^2C de datos (SDA) y que se organizan de la siguiente forma:

- El primer bit se corresponde con la señal de Inicio de la comunicación. Esta se define como una bajada del nivel lógico de alto a bajo mientras la señal de reloj se encuentre en nivel alto.
- Tras el bit de Inicio, se envía, en binario, la dirección del multiplexor PCA9540BD, con el que se pretende establecer comunicación.
- El tercer paquete consta de un único bit que indica si la acción requerida por el maestro es recibir información (1) o enviársela al esclavo (0). En el caso del sistema desarrollado, la placa Arduino pretende obtener la información de los sensores por lo que la acción será siempre de lectura.
- El siguiente bit es un bit de reconocimiento por parte del esclavo (multiplexor) para certificar que la comunicación ha sido establecida correctamente. Este se corresponde con una bajada del estado lógico de alto a bajo por parte del multiplexor.

- Una vez establecida la conexión con este, el siguiente paquete formado por 8 bits es el llamado registro de control. Este paquete tiene la función de seleccionar el canal del multiplexor con el que se pretende establecer conexión (sensor 1 o sensor 2). La selección del canal se realiza mediante los dos últimos bits de este grupo.
- El número de bytes (grupos de 8 bits) de datos transferidos del transmisor (esclavo) al receptor (maestro) entre la señal de Inicio y la señal de Parada es ilimitado. Tras el envío de cada byte de información, el transmisor debe añadir un bit de reconocimiento.
- La comunicación se termina con una señal de Parada por parte del maestro. Esto es, una subida del estado lógico de bajo a alto.

Una vez explicado el protocolo de comunicación con los sensores por medio del PCA9540BD, se detallan a continuación los pines que componen al multiplexor en la *Figura 2.7*.

En ella se pueden ver los pines 1 y 2 que se conectarán al Arduino para su conexión I^2C . Los pines 3 y 6 se corresponden con la alimentación y tierra respectivamente. Por último, los pines SCX y SDX hacen referencia a las salidas de señal de reloj y datos para la comunicación con los canales 0 y 1 que se conectarán a los sensores de presión diferencial. Todas estas conexiones se realizarán por medio de la PCB diseñada, explicada en la sección 2.2.5.

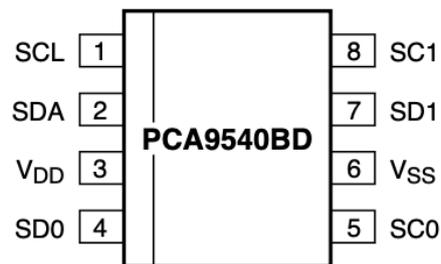


Figura 2.7: Pinout del multiplexor PCA9540BD

Además, para mantener las líneas en alto mientras no se lleve a cabo la comunicación, se añaden a los pines conectados al Arduino y a los sensores resistencias de pull-up siguiendo las indicaciones de la ficha técnica del componente.

La conexión completa mediante el bus I^2C entre el Arduino, el multiplexor y los sensores se puede observar en el esquemático de la *Figura 2.10*, en la sección 2.2.3.

2.2.3. Sensores MS4525DO

Como se ha mencionado en repetidas ocasiones, el objetivo del presente Trabajo es el diseño de un sistema para medir el ángulo de ataque de una aeronave. Para ello se hace uso de dos sensores de presión diferencial capaces de medir las variaciones de presión respecto de la atmosférica debido al término de presión dinámica explicado de la *Ecuación 1.1*. La función esencial que desempeñan hace que la correcta elección de los mismos resulte de gran importancia para el desarrollo del Trabajo.

El primer aspecto a tener en cuenta es la precisión requerida por el sensor. En este sentido, se tiene en cuenta que la diferencia entre un ala en condiciones de pérdida y de una que no lo está es del orden de 1° . De manera similar, para notar los beneficios de volar según la velocidad de máximo alcance o de mínima potencia requerida, el ángulo de ataque debe encontrarse en torno a $\pm 1^\circ$ del óptimo en cada caso.

En segundo lugar, es importante determinar el rango de valores de presión en los que debe trabajar el sistema. Para ello se presenta la *Tabla 2.1* donde se recogen las presiones dinámicas para distintas condiciones del vuelo de una aeronave. Los valores de la velocidad se corresponden con el de entrada en pérdida y crucero para una aeronave ligera.

Altura [m]	ρ [kg/m ³]	Velocidad [m/s]	P_{Din} [Pa]	P_{Din} [psi]
0	1.225	26	306	0.06
0	1.225	102	6484	0.94
2000	1.006	26	332	0.05
2000	1.006	102	5325	0.77

Tabla 2.1: Rango de presiones para distintas condiciones de vuelo

De los valores expuestos en la tabla anterior, y tras analizar los rangos de presión ofrecidos por los sensores disponibles en el mercado, se concluye que para el desarrollo del Trabajo se requerirá un rango de 0-1 psi (0-6894 Pa). Teniendo en cuenta la precisión requerida del orden de 1° en la medida y usando como punto de partida la *Figura 1.4* mostrada en la sección de *Justificación 1.1*, los sensores deben ser capaces de notar variaciones del orden de 0.01 psi, por lo que el error del mismo no deberá ser mayor al 1%.

Una vez se han fijado los requisitos de rango y de error de los dispositivos a implementar, se muestra a continuación, en la *Tabla 2.2*, una serie de modelos de sensores de presión diferencial existentes en el mercado que se barajaron para su posible implementación al sistema.

Sensor	Tensión de alimentación requerida [V]	Rango de presión [psi]	Rango de temperatura compensada [°C]	Error	Precio [€]
MPX2010DP	10	[0, 1.45]	-	1 %	13
MPX5010DP	5	[0, 1.45]	-	5 %	15
SSCDRRN001PD2A5	5	[0, 1]	[-20, 85]	2 %	31
HSCDRRN001PD2A3	3.3	[0, 1]	[0, 85]	1 %	45

Tabla 2.2: Resumen de los sensores de presión buscados para implementar en el sistema

Finalmente, con el objetivo de cumplir todos los requisitos y mejorar la precisión todo

lo posible sin elevar el coste del sistema, se optó por escoger el modelo de *Measurement Specialties* MS4525DO (ver *Figura 2.8*), más concretamente la versión MS4525DO-DS-5-A-I-001-D-P.

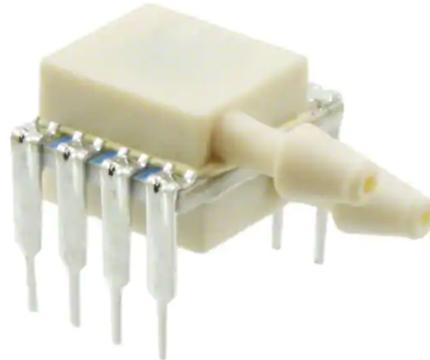


Figura 2.8: Sensor de presión diferencial MS4525DO

Se trata de un sensor de presión diferencial pequeño basado en material cerámico. El modelo específico del Trabajo posee dos tubos de entrada y tiene pines pasantes para su implementación en el sistema. Además, entre sus principales características destacan las mostradas en la *Tabla 2.3*:

Característica	Detalle
Tensión de entrada	5 V
Corriente de salida	3 mA
Rango de presión	[0, 1] psi
Error del sensor	0.25 %
Rango de temperatura compensada	[-10, 85] °C
Rango de temperatura de operación	[-25, 110] °C
Comunicación	I^2C
Masa	3 g

Tabla 2.3: Características del MS4525DO-DS-5-A-001-D-P

Como se ha mencionado, los sensores elegidos cuentan con dos tubos de entrada para realizar la medida de presión. Puesto que se pretende obtener de cada uno de los sensores el valor de la presión dinámica, uno de los tubos de cada uno de ellos estará cerrado al valor de la presión atmosférica, mientras el otro medirá la presión total (P_{Tot} según la *Ecuación 1.1*). Para ello, se conecta al puerto 1 (según la ficha técnica de los sensores disponible en el *Anexo A*), en consonancia con los resultados obtenidos con esta disposición en el documento [6], un tubo de silicona, dejando el puerto 2 abierto al aire de la cabina. Como se ha explicado en la sección de *Justificación 1.1*, el sensor posee una membrana interna que ofrece directamente la diferencia entre estas dos presiones. En el caso del sensor seleccionado, esta medida está comprendida en un rango que va de 0 a 16383 cuentas. Y esta medida se corresponde con la presión dinámica de cada sensor.

Una vez se han descrito sus características principales, se procede a explicar la conexión I^2C entre el multiplexor PCA9540BD y ambos sensores. Con este objetivo se muestra en la *Figura 2.9* el esquema de los pines del MS4525DO. En ella se pueden ver los pines de alimentación y tierra, que irán conectados entre ellos por un condensador de 100 nF según

las especificaciones de la ficha técnica. También se pueden ver los pines de la entrada de la señal de reloj (SCL) y la de datos (SDA). Estas dos entradas serán las que se conecten al multiplexor, a las salidas de los canales 0 y 1.

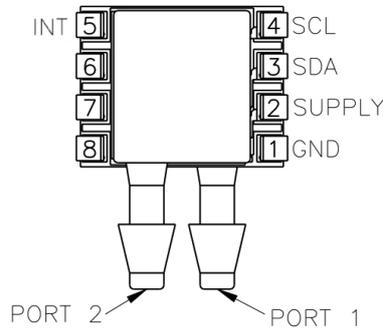


Figura 2.9: Pinout del sensor de presión diferencial MS4525DO

Añadiendo a la comunicación explicada entre el Arduino UNO y el multiplexor, las existentes entre este último y los dos sensores, la conexión total de estos componentes del sistema queda representada en la *Figura 2.10*.

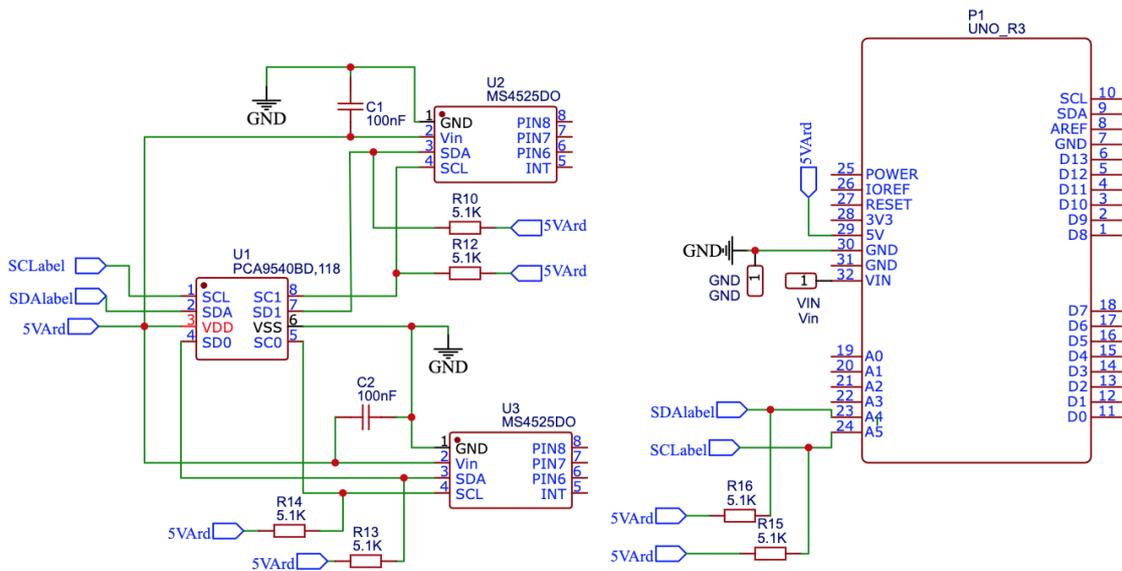


Figura 2.10: Esquemático conexión entre el Arduino y los sensores

Se pueden observar en ella, además de los componentes principales, las resistencias de pull-up de $5.1\text{ k}\Omega$ empleadas en las líneas del bus I^2C , así como los condensadores previamente citados. También se ve cómo la alimentación de todos los componentes proviene del pin de 5 V del Arduino UNO al que se encuentran conectados.

2.2.4. Pantalla ILI9341

En esta sección se explican las características principales de la pantalla elegida para el sistema y se detalla también la interfaz que permite mostrar el valor del ángulo de ataque.

La pantalla gráfica LCD (Liquid-Crystal Display) seleccionada tiene un tamaño de 2.8 pulgadas en diagonal y una resolución de 320 píxeles de ancho y 240 de alto (ver *Figura 2.11*). Además, permite mostrar información de texto y gráficos a color.

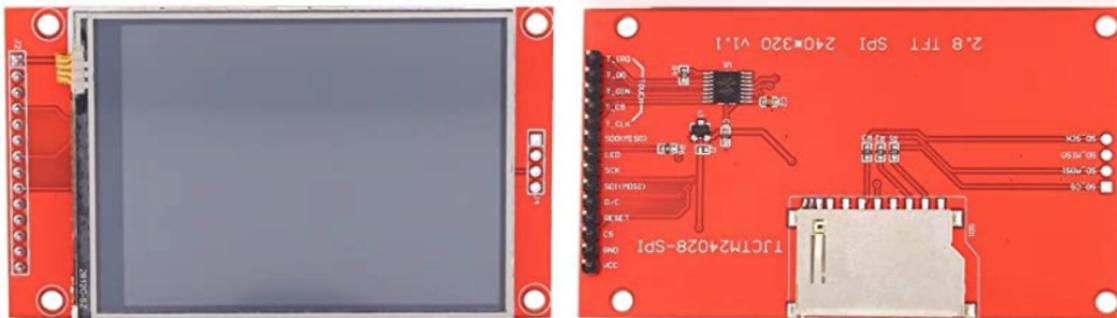


Figura 2.11: Pantalla gráfica LCD de 2.8 pulgadas

Está basada en el controlador ILI9341 y, en el caso de la empleada en este sistema, no posee capacidad táctil. Cuenta con una ranura para la inserción de una tarjeta SD aunque no se hará uso de ella.

La pantalla también cuenta con un regulador lineal de tensión de 5 a 3.3 V. Esto permite conectarla directamente, al igual que el resto de componentes, al pin de salida de 5 V del Arduino UNO.

Sin embargo, se usan puentes resistivos para bajar los 5 V de los pines de salida del Arduino a los 3.3 V de deben recibir los pines de datos de la pantalla. Para ello se emplean resistencias de valor de 1 k Ω como se puede ver en la *Figura 2.12*.

Como se ha mencionado en la sección 2.2.1, la comunicación entre la pantalla y la placa maestro se realiza mediante el bus SPI. La *Tabla 2.4* muestra en detalle las conexiones entre los pines de ambos elementos.

Pin Arudino	Pin Pantalla
13	SCK
11	MOSI
10	CS
9	D/C
5 V	LED - RST - VCC

Tabla 2.4: Pines de conexión entre el Arduino y la pantalla

Recordando lo explicado sobre la comunicación SPI, la señal entrante por el pin SCK se corresponde con la de reloj enviada por el maestro, el pin MOSI se encarga de la transmisión de datos del Arduino a la pantalla y CS corresponde con el cable de selección para iniciar la comunicación con la misma. Por su parte, los pines de LED, D/C, reset y alimentación son necesarios para su funcionamiento.

Una vez explicadas todas las conexiones existentes entre ambos dispositivos, se presenta el esquemático resultante en la *Figura 2.12*. Como se puede ver, en ella se representan las uniones entre los pines comentados, incluyendo las resistencias empleadas para obtener las tensiones deseadas. Además de puede observar cómo se dejan libres los pines destinados al uso de la tarjeta SD y el pin MISO que no se requiere en esta comunicación.

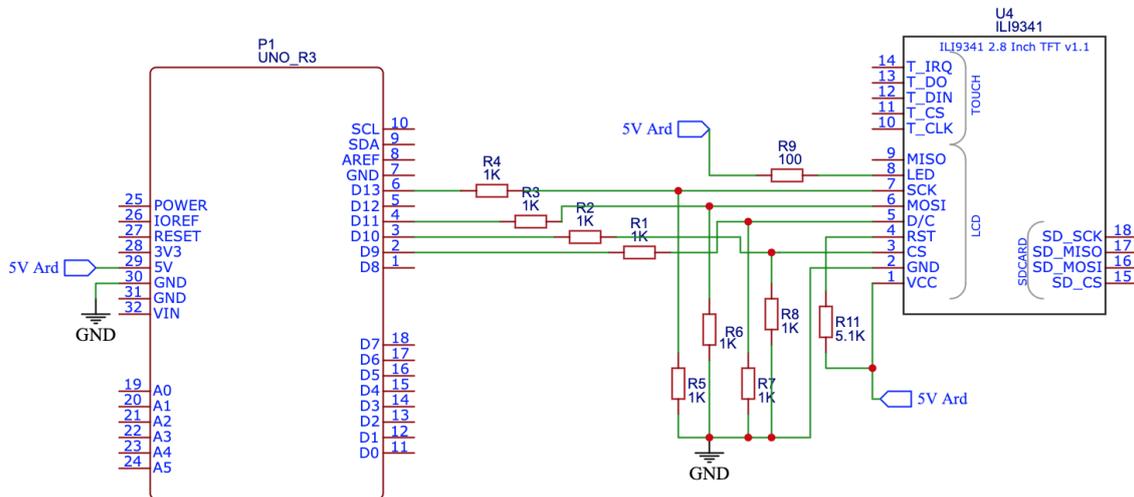


Figura 2.12: Esquemático de la conexión entre el Arduino y la pantalla

Por otra parte, se pretende explicar también en esta sección la interfaz creada para mostrar la información. Como se ha mencionado anteriormente, el objetivo es que esta sea sencilla y rápida de asimilar por lo que se opta por añadir al valor numérico del ángulo medido una pequeña representación en color del posible peligro que este pueda conllevar. Se trata de la silueta de un perfil aerodinámico que adquiere mayor inclinación según un rango preestablecido de valores y cambia, en consonancia, de color. Se emplean los colores de un semáforo para mayor simplicidad, esto es: verde, amarillo y rojo, siendo este último el relacionado con un mayor ángulo de ataque. La *Figura 2.13* muestra la interfaz recién explicada en sus tres estados, con un pantallazo de cada uno.



Figura 2.13: Interfaz de la pantalla con la información del ángulo de ataque

2.2.5. PCB

Para acabar con el apartado de hardware, se destina esta sección a la descripción de la placa de circuito impreso o PCB diseñada. Su implementación en el sistema tiene por objetivo principal evitar conexiones aparatosas mediante cables y componentes que dificulten el ensamblado del prototipo.

Una PCB es una superficie constituida por caminos y pistas de material conductor laminadas sobre una base no conductora. El circuito impreso se utiliza para conectar eléctricamente (a través de las pistas conductoras) y sostener mecánicamente (por medio de la base) un conjunto de componentes electrónicos. Las pistas son generalmente de cobre, mientras que la base se fabrica generalmente de resinas o polímeros.

Para el desarrollo de la misma, se ha empleado la página web de EasyEDA [11] que permite el diseño de la placa ofreciendo una serie de herramientas útiles para esta labor.

En primer lugar se crea el esquemático del sistema completo, formado por todas las conexiones y componentes necesarios para su funcionamiento. A lo largo de la sección *2 Descripción del sistema* se ha visto dicho esquema por partes y se presenta ahora, en la *Figura 2.14*, el diagrama completo.

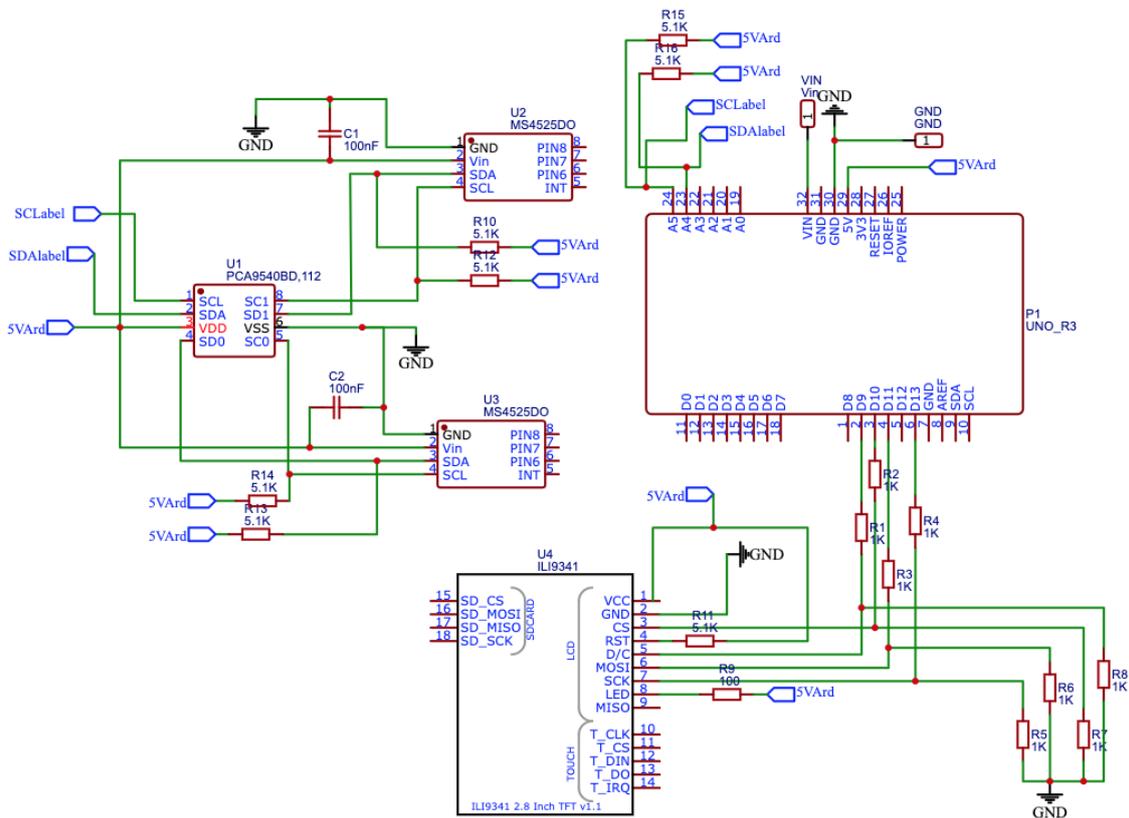


Figura 2.14: Esquemático completo del sistema, utilizado para el diseño de la PCB

A partir de este, la página crea una interfaz donde se realiza el diseño propiamente dicho de los caminos, las pistas, las láminas y se llevan a cabo todas las conexiones que muestra el esquemático.

En este punto cabe destacar que la placa creada cuenta con dos niveles de capas de pistas de cobre, así como de una capa especial a la que se conectan todas las tomas de tierra que las conexiones precisan. El resultado del proceso de diseño de la PCB con las pistas comentadas se muestra en la *Figura 2.15* donde se muestran conexiones en las dos capas: roja y azul.

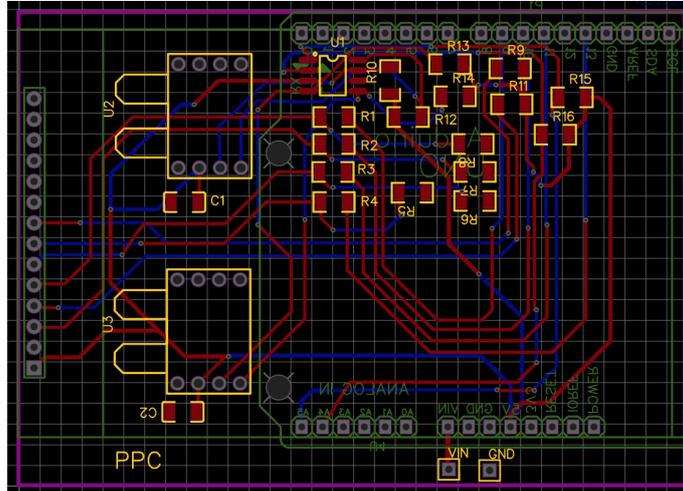


Figura 2.15: Detalle del diseño de la PCB con las pistas creadas para las conexiones

Respecto al cambio entre las dos capas comentadas en el párrafo anterior, este se realiza por medio de vías que están representadas por pequeños círculos en la PCB y que permiten la conexión entre ellas. También cabe mencionar que una de las cuestiones importantes de las pistas tiene que ver con que los giros se realicen por medio de ángulos de 45° , permitiendo así el flujo de electricidad de manera eficaz.

El procesado de la placa se realizó mediante la empresa JLCPCB, que llevan a cabo tareas de recorte, taladrado, añadido de las capas de cobre, tests eléctricos, etc. que aseguran su correcto funcionamiento.

Así pues, la *Figura 2.16* muestra el resultado final del proceso, al cual se sueldan todos los componentes que conforman el sistema (Arduino UNO, pantalla, multiplexor, sensores, resistencias y condensadores), como se puede ver por sus siluetas en el grabado de la misma.

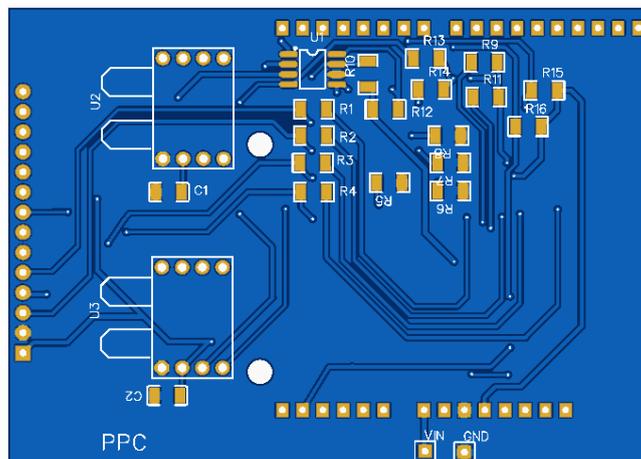


Figura 2.16: PCB diseñada para el montaje de los componentes del sistema

Como se mencionó en la sección 2.1 *Diagrama de componentes*, a la PCB se suelda en una de las caras la pantalla, mientras que en la otra se encuentran el resto de componentes. Esta manera de montaje permite que el usuario pueda centrar su atención únicamente en la pantalla, que ofrece la información requerida. Además, la estructura propuesta reduce el espacio que el sistema necesita para su implementación. Esto se puede ver en las *Figuras 2.17 y 2.18* que muestran el resultado final una vez soldados todos los componentes.



Figura 2.17: Estructura final del sistema montado sobre la PCB

Se puede observar en ella cómo la parte de alimentación de la placa Arduino, que sirve de alimentación al resto de componentes, queda al descubierto, facilitando su acceso para la conexión a una batería portátil mediante el cable USB-B ya mencionado en secciones anteriores. También se puede ver que se añaden pines macho/hembra para permitir el montaje y desmontaje de los elementos principales como son la pantalla y el Arduino.



Figura 2.18: Dimensión del sistema montado sobre la PCB

Como se aprecia, el ancho del sistema no supera los 4 centímetros de grosor, facilitando su implementación sin el requerimiento de una gran cantidad de espacio para ello.

2.3. Software

Hasta ahora se han explicado la estructura completa del sistema diseñado por medio de un análisis del hardware empleado, así como del montaje llevado a cabo para su implementación. Se pretende en esta sección terminar con la explicación del sistema, haciendo énfasis, en este caso, en el software utilizado.

Se divide esta sección en dos partes. En primer lugar se explica brevemente la plataforma IDE de Arduino en la que se ha desarrollado el programa a ejecutar por el sistema. Tras ello, se explica la estructura de dicho código que permite la correcta toma de datos por parte de los sensores, el procesado de los mismos por la placa Arduino UNO y su envío a la pantalla donde pueden ser leídos a través de la interfaz ya comentada.

2.3.1. IDE Arduino

El entorno de desarrollo integrado también llamado IDE (siglas en inglés de Integrated Development Environment), es un programa informático compuesto por un conjunto de herramientas básicas de programación. Estas consisten en un editor de código, un compilador, un depurador, gestor de librerías y un constructor de interfaz gráfica (GUI). Además en el caso de Arduino permite cargar el programa ya compilado en la memoria flash del hardware.

Otra de las herramientas útiles con las que cuenta el IDE de Arduino es un monitor serie. Este permite visualizar por pantalla los valores que el Arduino almacena durante la ejecución del código. En el caso particular de este sistema, esto será de especial relevancia a la hora de poder mostrar los valores que se reciben tras su comunicación con los sensores, por ejemplo a la hora de calibrarlos así como de comprobar que las lecturas realizadas se encuentran dentro del rango esperado.

Cabe destacar también que este entorno de programación se encuentra disponible tanto para Windows, Linux como MacOS X. Además, la conexión entre el Arduino y el ordenador se realiza de forma muy sencilla, a través del ya comentado puerto USB-B del que dispone la placa, de manera que el propio ordenador sirve como alimentación durante la subida del código al Arduino.

En la *Figura 2.19* se muestra el entorno IDE explicado donde se realiza el desarrollo del código informático y su subida a la placa Arduino UNO para su ejecución en el sistema.



Figura 2.19: Entorno IDE de Arduino

2.3.2. Código

Se presenta a continuación el código al completo con el que se programa la placa Arduino UNO, encargada de requerir y enviar la información necesaria para el funcionamiento del conjunto del sistema.

Como se verá, el código se encuentra comentado, sin embargo se explican brevemente, previo a su visualización, las funciones generales llevadas a cabo por el mismo.

En primer lugar, se incluyen las librerías necesarias que facilitan los comandos a utilizar para generar el código. Como se puede ver, se emplean las librerías *SPI.h* y *Wire.h* que permiten realizar las comunicaciones SPI e I^2C con la pantalla y los sensores respectivamente. También se añaden las librerías *Adafruit_GFX.h* y *Adafruit_ILI9341.h* que se usan para el manejo de los gráficos de la pantalla LCD.

A continuación se definen una serie de puertos referentes a la pantalla así como las direcciones I^2C del multiplexor y de los sensores. Junto con esto se inicializan todas las variables de las que hace uso el programa para calcular el valor final del ángulo de ataque que se mostrará por la pantalla junto a la representación de los perfiles alares correspondientes.

Tras esto, se encuentra el llamado *void setup()*. En este se inicializan los parámetros de la comunicación en serie I^2C , se genera el entorno de la interfaz de la pantalla sobre la que se actualizarán las medidas del ángulo de ataque y se cargan las constantes de la recta de calibración cuya obtención se explica en la sección 3 *Calibración*.

Así se llega al bucle principal que se estructura de la siguiente manera:

- Se selecciona el canal del multiplexor correspondiente al primer sensor mediante el protocolo I^2C explicado en la sección 2.2.2 *Multiplexor PCA9540BD*.
- Se obtiene la presión del sensor seleccionado.
- Se repite el proceso con el segundo sensor.
- Se corrigen los valores leídos de presión dinámica de ambos sensores y se obtiene el ratio entre ellos.
- Se emplea la ecuación obtenida mediante la calibración del sistema para obtener el ángulo de ataque correspondiente a los valores obtenidos.
- Se realiza la comunicación SPI con la pantalla que muestra el resultado al usuario.

Cabe destacar que tanto para la selección del canal del multiplexor como para la obtención de la presión de los sensores, se crean las funciones *selec_canal(canal)* y *medirPresMS4525DO(sensor)* que se muestran al final del código y facilitan su desarrollo y comprensión.

```

1  /*
2  Codigo para implementar en el sistema de medida
3  del angulo de ataque mediante dos sensores de presion
4  diferencial MS4525DO-DS-5-A-I-001-DP y enviado a pantalla
5  LCD ILI9341
6
7  Autor: Pablo Perez Correa
8
9  */
10
11 #include <SPI.h>           // incluye libreria bus SPI
12 #include <Wire.h>         // incluye libreria comunicacion en
    serie I2C
13 #include <Adafruit_GFX.h> // incluye libreria para manejo de
    graficos
14 #include <Adafruit_ILI9341.h> // incluye libreria para controlador
    ILI9341
15
16
17 #define TFT_DC 9          // define constante TFT_DC con numero 9 (
    puerto del Arduino)
18 #define TFT_CS 10        // define constante TFT_CS con numero 10 (
    puerto del Arduino)
19 #define GREEN 0x07E0      // define constante el color VERDE
20 #define YELLOW 0xFFE0    // define constante el color AMARILLO
21 #define RED 0xF800        // define constante el color ROJO
22 #define Mult_Direc 0x70  // define direccion del multiplexor 1110000
    en binario
23 #define sensor_direc 0x28 // define direccion I2C del sensor 40 en
    decimal
24
25 Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC); // crea
    objeto para controlar pantalla
26
27 //VARIABLES//
28
29 // Variables comunicacion con el sensor
30 int Alfaround;           // valor redondeado del angulo de
    ataque
31 int elec;                // variable para seleccionar el canal
    0, 1 del multiplexor
32 int sensorfwd = 4;       // sensor fwd conectado al canal 0
    del multiplexor 'control register'
33 int sensor45 = 5;        // sensor 45 conectado al canal 1 del
    multiplexor 'control register'
34 int sensor;              // variable empleada en la funcion
    para obtener la presion
35 float deltapres;         // presion medida por un sensor
36 float A, B, C;           // constantes de la recta de
    calibracion
37 float pfwdnocarga, p45nocarga; // valor de 'offset' de cada uno de
    los sensores
38 float pfwdcorr, p45corr; // resta entre el valor medido y el '
    offset'
39 float pfwd_p45, Alfa;    // ratio entre las presiones
    dinamicas. Angulo de ataque
40 int lectura = 0;         // lectura de los sensores (14 bits)
41 int valor = 63;          // 00111111 para comprobar longitud
    de MSB
42
43
44 // Variables de la pantalla
45 extern uint8_t AHorBM[]; // airfoil verde

```

```

46  extern uint8_t AOkayBM[]; // airfoil amarillo
47  extern uint8_t AStallBM[]; // airfoil rojo
48  long check; // variable de comprobacion del estado
    anterior
49
50  void setup(){ // Configuracion inicial
51  Serial.begin(9600); // inicio de la comunicacion serie
52  Wire.begin(); // se inicializa el protocolo I2C
53
54  // Valores que se deben obtener mediante la calibracion se deben
55  // ajustar para un caso concreto de localizacion de la sonda
56
57  // Alfa = A*(Pfwd/P45)^2 + B*(Pfwd/P45) + C
58  A = 4.3593;
59  B = -30.189;
60  C = 50.433;
61  pfwdnocarga = 8157; // valor del offset en
62  p45nocarga = 8156; // cuentas 0-16383 (14 bits)
63
64
65
66  tft.begin(); // inicializa pantalla
67  tft.setRotation(1); // establece posici n
    horizontal
68  tft.fillScreen(ILI9341_BLACK); // fondo de pantalla de
    color negro
69  tft.fillRect(0, 0, 320, 45, ILI9341_NAVY); // rectangulo azul naval
70 // a modo de fondo de
    titulo
71  tft.setTextColor(ILI9341_WHITE); // color de texto en
    blanco
72  tft.setTextSize(3); // escala de texto en 3
    (18 x 21 pxl)
73  tft.setCursor(25,11); // ubica cursor
74  tft.print("Angle of attack"); // imprime texto
75 }
76
77  void loop(){
78
79  // OBTENCION DE PRESIONES Y ANGULO DE ATAQUE MEDIANTE SENSORES//
80
81  // OBTENCION DE LAS PRESIONES
82  elec=sensorfwd;
83  selec_canal(elec); // seleccion del sensor que se
    multiplexa, en este caso, Pfwd
84  float pfwd = medirPresMS4525DO(elec); // obtencion de la presion
    dinamica del sensor de Pfwd
85
86
87  pfwdcorr = pfwd - pfwdnocarga; // presion corregida con el 'offset'
    de Pfwd
88
89  elec=sensor45;
90  selec_canal(elec); // seleccion del sensor que se
    multiplexa, en este caso, P45
91  float p45 = medirPresMS4525DO(elec); // obtencion de la presion
    dinamica del sensor de P45
92
93  p45corr = p45 - p45nocarga; // presion dinamica de P45
94
95  Serial.flush(); // limpia el puerto de comunicacion
96
97  pfwd_p45 = pfwdcorr/p45corr; // calculo del ratio entre ambas

```

```

    presiones dinamicas obtenidas
98
99 // OBTENCION DEL ANGULO DE ATAQUE
100 Alfa = A*(p45)*(p45) + B*(p45) + C;
101
102 Alfaround=round(Alfa); // redondea el valor a la unidad m s proxima
    para mostrar en pantalla
103
104
105 // SALIDA PUERTO SERIE
106 Serial.print(p45); // comprobacion de normalidad y visualizacion
    de los datos
107 Serial.print(" ");
108 Serial.print(p45corr); // comprobacion de normalidad y
    visualizacion de los datos
109 Serial.print(" ");
110 Serial.print(p45); // comprobacion de normalidad y visualizacion de
    los datos
111 Serial.print(" ");
112 Serial.print(p45corr); // comprobacion de normalidad y
    visualizacion de los datos
113 Serial.print(" ");
114 Serial.print(p45); // comprobacion de normalidad y
    visualizacion de los datos
115 Serial.print(" ");
116 Serial.print(Alfaround); // comprobacion de normalidad y
    visualizacion de los datos
117 Serial.println(" ");
118
119
120 // DISPLAY EN PANTALLA
121
122 tft.setTextColor(ILI9341.WHITE, ILI9341.BLACK); // texto en color
    blanco con fondo negro
123 tft.setTextSize(10); // escala de texto en
    10
124
125 if(Alfa>=10){
126     tft.setCursor(188,103); // ubica cursor
127     tft.print(Alfaround); // valor que representa el angulo de
    ataque medido y calculado por los sensores
128 }
129 else{
130     tft.setCursor(188,103); // ubica cursor
131     tft.print(" ");
132     tft.print(Alfaround); // valor que representa el angulo de
    ataque medido y calculado por los sensores
133
134 // Dependiendo del valor de Alfaround se muestra un airfoil u otro
135 // con mayor o menor inclinacion y su correspondiente color
136
137 }
138 if(Alfa>=12 && check!=1){
139     tft.fillRect(10, 70, 160, 140, ILI9341.BLACK);
140     drawBitmap(10, 70, AStallBM, 160, 140, RED);
141     check=1;
142 }
143 else if(Alfa>=6 && Alfa<12 && check!=2){
144     tft.fillRect(10, 70, 160, 140, ILI9341.BLACK);
145     drawBitmap(10, 70, AOkayBM, 160, 140, YELLOW);
146     check=2;
147 }
148 else if(Alfa<6 && check!=3){

```

```

149     tft.fillRect(10, 70, 160, 140, ILI9341.BLACK);
150     drawBitmap(10, 70, AHorBM, 160, 140, GREEN);
151     check=3;
152 }
153 delay(1000);           // se establece una demora de 1000 mseg. aunque
154                       // este valor puede variar en funci n de si nos
155     encontramos      // en la fase de calibracion y pretendemos obtener
156     valores          // cada mas tiempo o menos
157 }
158
159
160 // FUNCIONES UTILIZADAS EN EL CODIGO
161
162 // FUNCION QUE INSERTA LA IMAGEN DE LOS PERFILES ALARES
163
164 void drawBitmap(int16_t x, int16_t y, const uint8_t *bitmap, int16_t w,
165               int16_t h, uint16_t color){
166     int16_t i, j, byteWidth = (w+7)/8;
167     uint8_t byte;
168     for(j=0; j<h; j++){
169         for(i=0; i<w; i++){
170             if(i & 7) byte <<= 1;
171             else     byte  = pgm_read_byte(bitmap + j * byteWidth + i / 8);
172             if(byte & 0x80) tft.drawPixel(x+i, y+j, color);
173         }
174     }
175 }
176
177
178 // FUNCION QUE SELECCIONA EL CANAL DEL PCA9540BD (MULTIPLEXOR) CON EL
179     QUE SE CONECTA EL ARDUINO
180
181 void selec_canal(uint8_t canal){
182     Wire.beginTransmission(Mult_Direc); // inicio de la comunicacion con
183     el multiplexor
184     Wire.write(canal);                  // se selecciona al canal 0 o 1
185     Wire.endTransmission();            // se espera el bit de
186     reconocimiento por parte del multiplexor
187 }
188
189 // FUNCION QUE LEE LA PRESION DINAMICA MEDIDA POR EL SENSOR MS4525DO
190
191 float medirPresMS4525DO(int sensor){
192     deltapres = 0;
193     Wire.beginTransmission(sensor_direc); // inicio de la comunicacion
194     con el sensor, ya multiplexado
195     Wire.write(0);                       // se pide acceder al registro
196     de datos
197     Wire.endTransmission();              // se completa la transmision
198     con el sensor
199
200     Wire.requestFrom(sensor_direc, 2); // se pide leer dos bytes, ya que
201     el sensor obtiene el valor de la presion en 14 bits
202     if(2 <=Wire.available())           // asegurar que los 2 bytes estan
203     listos para ser leidos
204     {
205         lectura = Wire.read();          // lee el primer byte, segun las
206         librerias, el MSB
207         lectura = lectura & valor;      // valor comprueba que la extension del

```

```
    MSB es la correcta, 6 bits
200  lectura = lectura << 8;    // se mueve el MSB a la izquierda 8
    posiciones
201  lectura |= Wire.read();    // lectura del byte 2. Se a ade al MSB
    desplazado, el LSB
202  deltapres = lectura;
203  }
204  return deltapres; // la funcion devuelve el valor de presion dinamica
    obtenido
205  }
```

2.4. Diseño sonda 3D

Para terminar con la parte de diseño del sistema, se explican a continuación los pasos seguidos para crear una sonda que permita conectar los tubos procedentes de los sensores (dentro de la cabina) con el aire incidente en el exterior del avión y facilite también la toma de datos para la calibración del sistema.

La pieza mencionada se ha diseñado en el programa Fusion 360 de Autodesk del cual se tiene licencia por ser alumno de la UPV.

Así pues, se pretende crear una pieza 3D similar a la configuración de un tubo de Pitot que se instalan en los aviones comerciales. En concreto, esta se localizaría en el intradós de una de las alas para el caso de este Trabajo. Además, para la aplicación concreta del método de Rogers que se explica en la *sección 3.1 Método de medición* se crean dos superficies en las que desembocarán los dos tubos procedentes de los sensores de presión, una vertical y la otra a 45° de esta.

Para ello se divide la pieza final deseada en dos, resultado de partir a esta por la mitad y, posteriormente, se unen ambas mitades. Esto se realiza de esta manera debido a que se necesita introducir por el interior de la pieza final los dos tubos de silicona que serán los encargados de conectar a los sensores con la corriente de aire incidente.

En primer lugar, para el desarrollo de una de las mencionadas mitades, se definen las medidas deseadas de la pieza y se generan geometrías básicas de esas medidas mediante la herramienta *Sketch* del programa. También se añaden las restricciones pertinentes como líneas paralelas, verticales, coincidentes, etc. Esto se puede ver en la *Figura 2.20*.

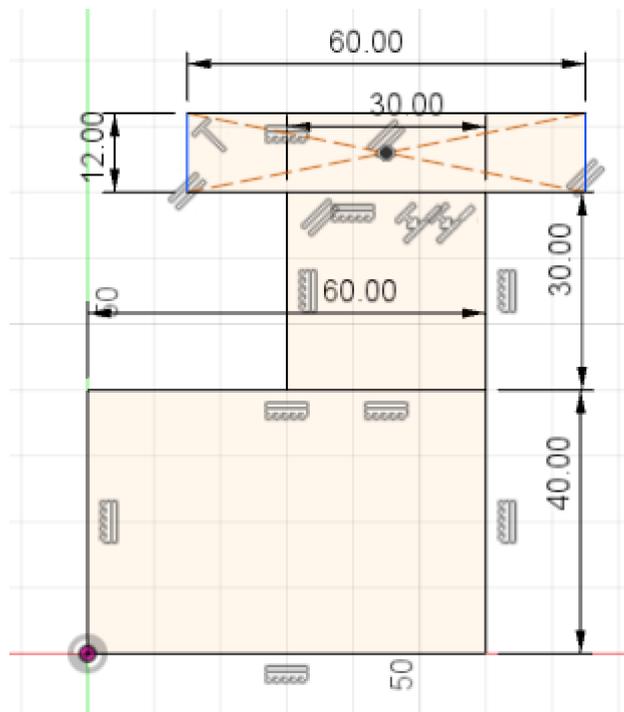


Figura 2.20: Sketch inicial para el diseño de la pieza 3D

A partir de dicha base, se realiza una extrusión de un grosor de 9 mm. También se redondean las esquinas mediante la opción *Fillet* con el objetivo de obtener una sección más aerodinámica que afecte en menor medida al flujo de aire que atraviesa la zona inferior del ala en la que se encuentra la sonda.

Como se puede observar, el sketch mostrado no posee todavía la superficie a 45° de la vertical. Esta se crea tras extruir la pieza, realizando una línea sobre la cara deseada y eliminando el material sobrante correspondiente a dicha esquina.

Para la creación de los surcos por los que circulan los tubos de silicona, se crea una línea que empieza en la parte superior de la pieza y desemboca en uno de los dos puertos mencionados. Tras esto, se crea una semicircunferencia con el diámetro de los tubos empleados (en este caso estos poseen un diámetro exterior de 4 mm y se dejan 0.2 mm de margen) en la cara superior y se emplea la herramienta *Sweep* que genera el recorrido a lo largo de la línea mencionada (ver *Figura 2.21*).

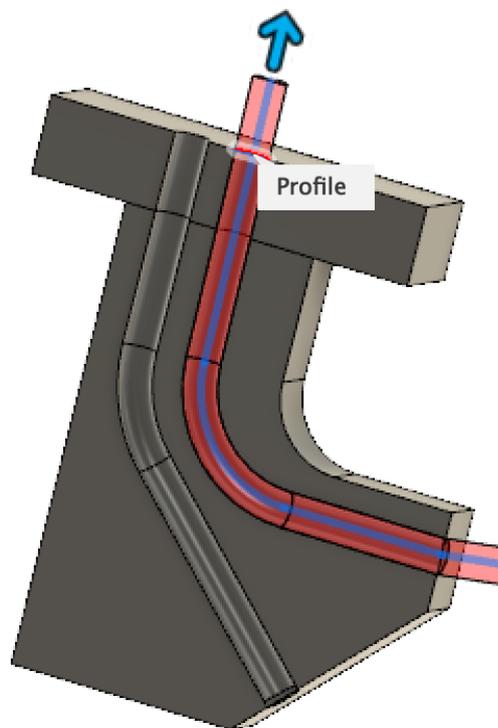


Figura 2.21: Imagen de la pieza con los surcos creados para los tubos de silicona

Antes de finalizar, se añade una zona más gruesa de material para reforzar la parte de unión entre el módulo superior y el brazo vertical de la pieza. También se crean los orificios necesarios para añadir los tornillos que sirvan de ayuda a la unión de las dos mitades, así como a su enganche a la aeronave.

Por último se crea el espejo de la pieza generada mediante la opción *Mirror*, dando por resultado el que muestra la *Figura 2.22*.

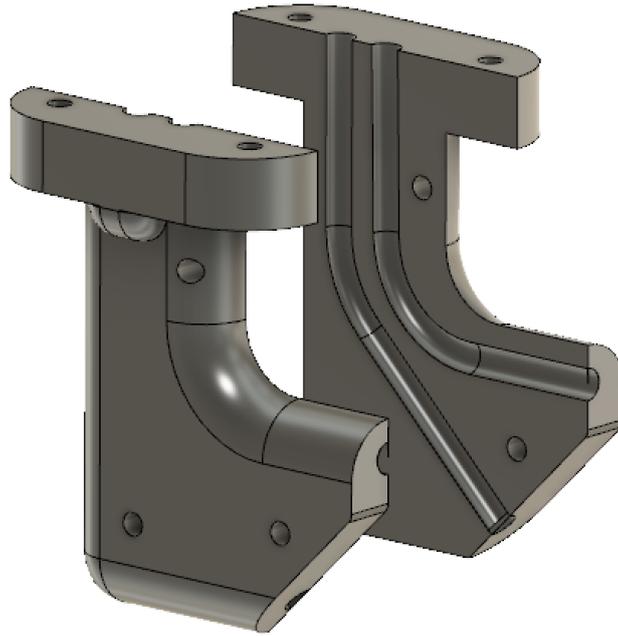


Figura 2.22: CAD final de la pieza 3D diseñada en Fusion 360

Una vez el diseño está finalizado, se exportan ambas mitades como archivos .STL independientes y se imprimen en una impresora 3D. El modelo concreto de la utilizada para este Trabajo es M3D micro y el material utilizado es PLA.

El resultado final de la pieza impresa y sellada mediante los tornillos de 3 mm se muestra en la *Figura 2.23*.



Figura 2.23: Pieza impresa en 3D y sellada con los tubos de silicona en su interior

Quedando así explicado el diseño completo del sistema.

3. Calibración del sistema

La calibración es un aspecto fundamental en el desarrollo de cualquier sensor. Esta consiste en comparar el valor que este mide con una referencia para asegurarse que se está midiendo correctamente (con unos márgenes de error aceptables).

En el caso del sistema al que hace referencia este Trabajo, y como se ha visto y se ha explicado en secciones anteriores, el microcontrolador requiere de la implementación de una función que relacione las presiones, medidas por ambos sensores, con el correspondiente valor del ángulo de ataque.

De esta manera, en esta sección se detallan los pasos realizados para asegurar que la medida ofrecida por el sistema al usuario coincide con la situación real del ángulo de ataque experimentado.

Para ello, en primer lugar se explica el método desarrollado por Rogers [5] con el que se pretende eliminar la dependencia de la medida con la velocidad, a la que, como se vio en la sección *1.3 Marco Teórico*, no se permite acceder.

Tras ello, se explica la calibración propiamente dicha. En este punto, cabe mencionar que por la situación derivada de la pandemia global de la Covid-19, no se ha podido realizar la parte experimental asociada a este apartado, ya que no se ha tenido acceso a las instalaciones del *Laboratorio Pedro Duque* en la Escuela Técnica Superior de Ingeniería del Diseño (ETSID) de la Universitat Politècnica de València (UPV) y, por tanto, al túnel de viento del que se pretendía disponer. En su lugar, se describe el proceso que se habría llevado a cabo de haber podido hacer uso del material mencionado y se muestra una calibración alternativa realizada; ya que el prototipo en su totalidad sí estaba montado y listo para su calibración, como se puede ver en la *Figura 3.1*.

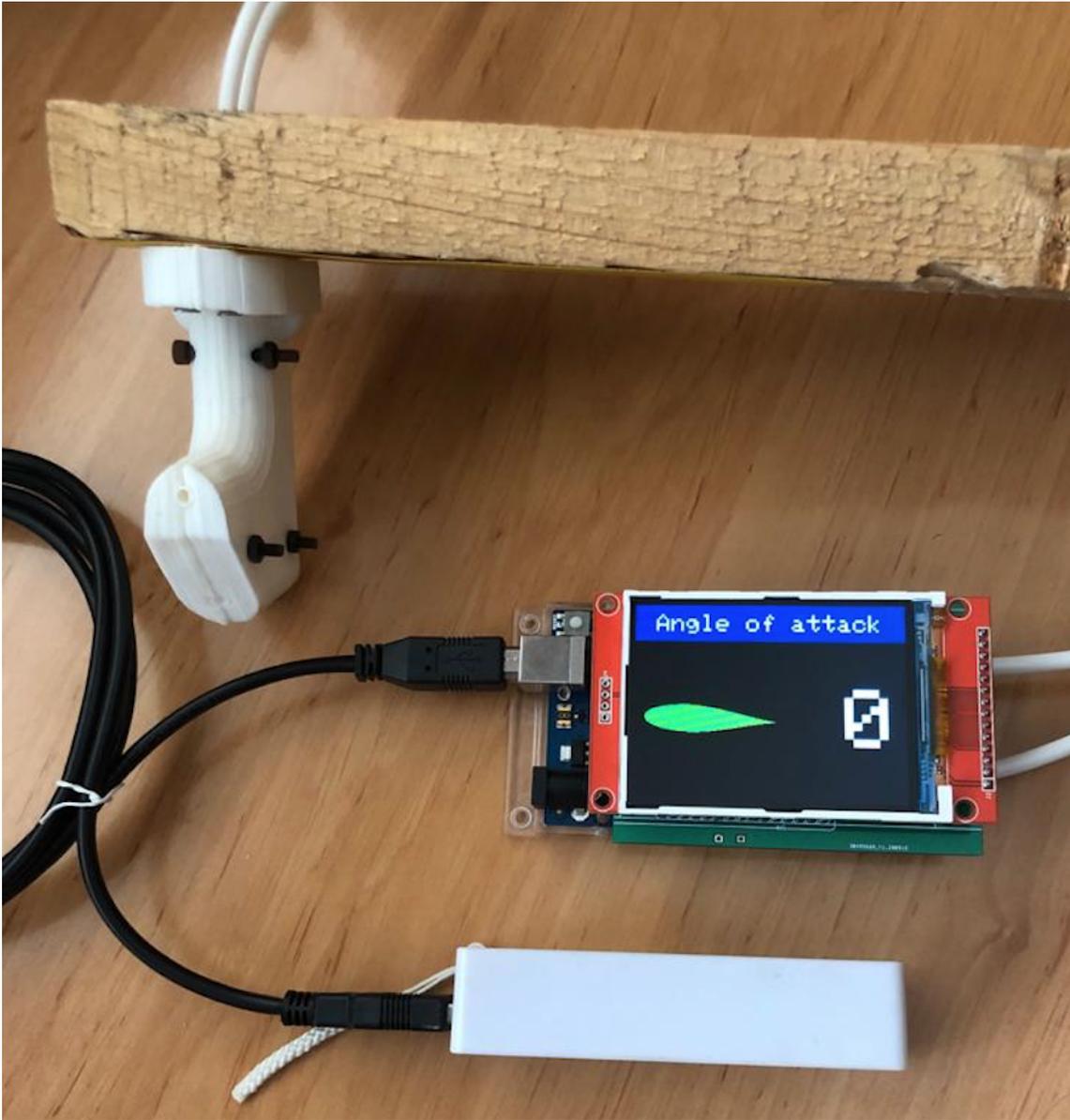


Figura 3.1: Sistema completo montado y listo para su calibración

3.1. Método de medición

Como se ha explicado, la FAA no permite que ningún dispositivo destinado a la medida del ángulo de ataque de una aeronave tenga acceso a sistemas ya certificados como son el indicador de velocidad o el tubo de Pitot de la misma.

Con el fin de evitar la dependencia respecto a la velocidad de las medidas de presión realizadas, se sigue el procedimiento llevado a cabo por David F. Rogers [5].

El método consiste en obtener la presión mediante dos puntos de una sonda que recibe el aire relativo sin perturbar. Esta sonda se suele colocar en la parte inferior del ala de la aeronave como se aprecia en la *Figura 3.2*. Los dos puntos mencionados de los que se pretende obtener el valor de la presión son P_{fwd} y P_{45} .

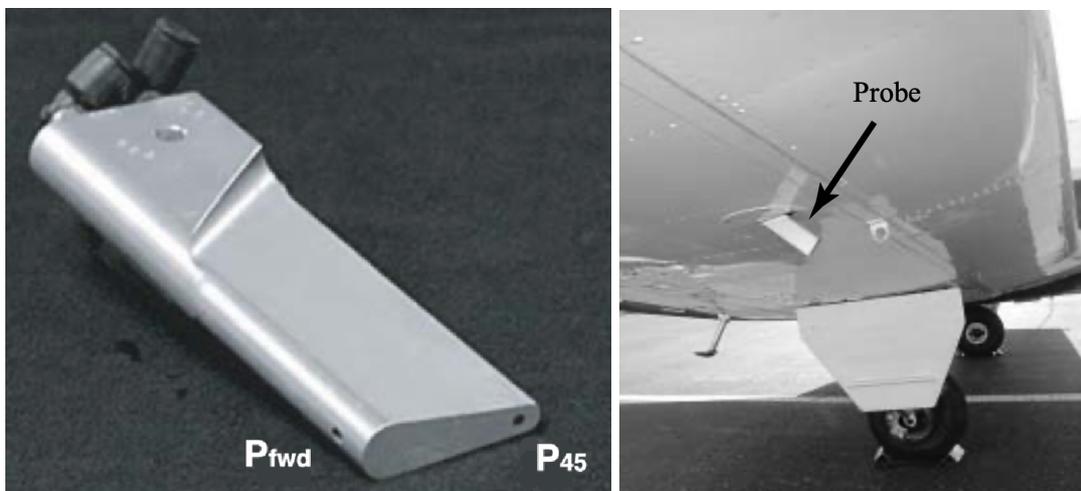


Figura 3.2: Sonda bajo el ala de una aeronave con dos puertos para medir P_{fwd} y P_{45}

El primero de ellos, P_{fwd} (presión delantera en inglés), se encuentra alineado con la cuerda del ala. De esta manera, en condiciones de crucero donde el ángulo de ataque es reducido, P_{fwd} mide el valor de la presión total que sufre la sonda.

Por su parte, P_{45} se encuentra localizado en la superficie plana de dicha sonda que forma un ángulo de 45 grados respecto de la superficie en la que se encuentra P_{fwd} .

Rogers, en su artículo, afirma que la diferencia de las presiones medidas en dichos puntos se relaciona con el ángulo de ataque de la aeronave. Sin embargo, realizando dicha resta, los valores obtenidos son dependientes de la presión dinámica y, por tanto, de la velocidad.

Para dar solución a este inconveniente, Rogers propone dividir la diferencia de las presiones obtenidas entre el valor de P_{45} . Al realizar esta operación, los puntos obtenidos para distintas presiones dinámicas (y por ende distintas velocidades y/o densidades), convergen todos en una misma curva. Esto se puede ver en la *Figura 3.3* donde se muestran los resultados de efectuar la operación explicada para cuatro valores de presión dinámica, q , diferentes.

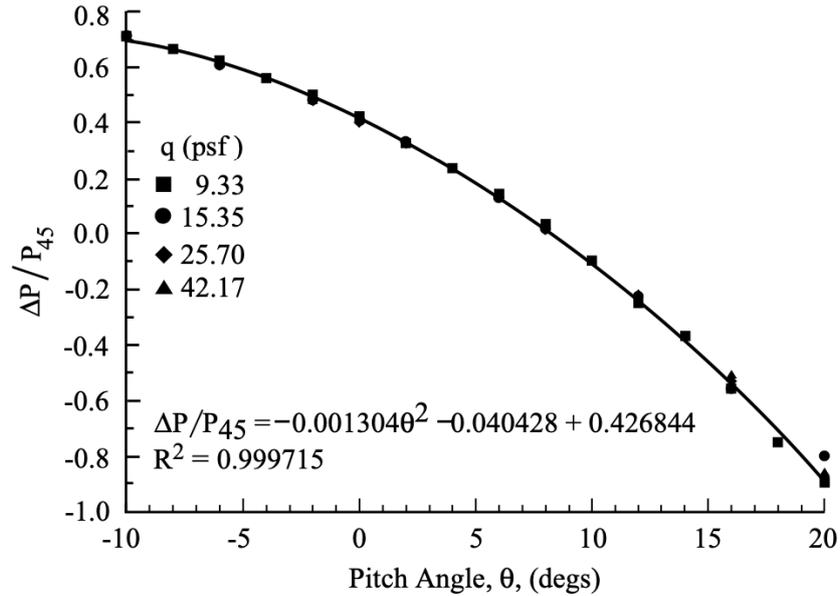


Figura 3.3: Curva normalizada de la relación entre las presiones y el correspondiente ángulo de ataque, independiente de la presión dinámica

En el documento citado [5], se indica que se obtiene un resultado similar si directamente se divide el valor de P_{fwd} entre el de P_{45} , es decir, independiente también de la velocidad incidente.

Este es el método que se emplea en este Trabajo para relacionar ambas variables. Uno de los sensores de presión diferencial explicados en 2.2.3 *Sensores MS4525DO* mide la presión dinámica del puerto denominado P_{fwd} mientras que el otro sensor se conecta al puerto localizado en la superficie plana a 45° del primero, para obtener el valor de P_{45} .

Cabe mencionar que Rogers realiza las medidas para varias condiciones de vuelo, combinando efectos de cabeceo, alabeo y guiñada. De estos experimentos, llega a la conclusión de que estas circunstancias afectan ligeramente al error del ángulo de ataque medido por el sistema aunque para rangos bajos (valores de crucero y hasta unos 15°) de valores de ángulos de guiñada y alabeo combinados con cabeceo, no se aprecian cambios significativos en la diferencia entre el valor medido y el valor real. Sin embargo, se deja como futuro trabajo a realizar el estudio en mayor profundidad de estos casos.

3.2. Resultados de la calibración

La calibración se pretendía llevar a cabo en el túnel de viento localizado en el *Laboratorio Pedro Duque* de la ETSID en el campus de Vera de la UPV. Sin embargo, como se ha explicado previamente, debido a las circunstancias a causa de la crisis de la Covid-19, no se ha tenido acceso a dichas instalaciones.

En su lugar, se ha realizado una calibración mediante un método alternativo. Para ello, se ha colocado la sonda 3D en el extremo de una tabla de madera y esta se ha montado en la parte trasera de un coche de manera que la tabla lo atravesaba de ventanilla a ventanilla como se puede ver en la *Figura 3.4*, sobresaliendo la sonda alrededor de 10 cm de esta. Además, para poder controlar el ángulo durante la medida se une a la tabla un teléfono móvil con una aplicación que ofrece el valor de inclinación del mismo.



Figura 3.4: Montaje para la calibración en carretera

Las pruebas se realizaron el día 18 de junio de 2020 en la Autovía del Mediterráneo en el tramo que une Valencia y Puzol. Se prestó especial atención a las medidas de seguridad pertinentes de manera que la tabla no invadiese el carril contiguo y para la obtención de los datos se empleó en todo momento el carril derecho donde no se interfirió en la circulación normal de ningún vehículo ajeno al experimento.

Para la realización de las medidas de calibración, se empleó un pequeño *script* en Python llamado *serialToExcel.py* que se puede encontrar en el ANEXO B. Este almacena los valores enviados de las presiones de P_{fwd} y P_{45} corregidas con el valor de offset, como se realiza en el código de la sección 2.3.2. De esta manera, la toma de datos se realiza de una forma sencilla y sistemática. Se programó de manera que se almacenaran los valores ofrecidos por ambos sensores cada 0.4 segundos durante 36 segundos, obteniendo un total de 90 parejas de valores (un dato de cada sensor) por medida.

Gracias a esta implementación, bastaba con conectar el sistema al ordenador mediante el cable USB-B y ejecutar el programa en Python. Al terminar el período de muestreo, el Excel se guarda con el nombre correspondiente para el posterior tratamiento de los datos obtenidos.

El procedimiento llevado a cabo fue análogo al que se hubiera realizado en el túnel de viento (adaptado a las condiciones disponibles). Se realizaron medidas para ángulos comprendidos entre 0 y 20 grados de inclinación de la sonda en intervalos de 5 grados. La medida se realizó para una velocidad constante de 100 km/h que se estableció con el controlador de velocidad del vehículo, siendo conscientes de que esta velocidad es inferior a la de entrada en pérdida de una aeronave. Sin embargo el límite de velocidad de la autovía, añadido al hecho de tener que realizar las medidas en el carril derecho, imposibilitaban velocidades mayores que sí se podrían haber tenido en cuenta en el túnel de viento.

Así pues, se generaron un total de 5 hojas de cálculo. Tras descartar algunos datos debidos a errores en la adquisición de la medida, se realizó el promedio de los valores para cada ángulo y se consiguió la siguiente gráfica.

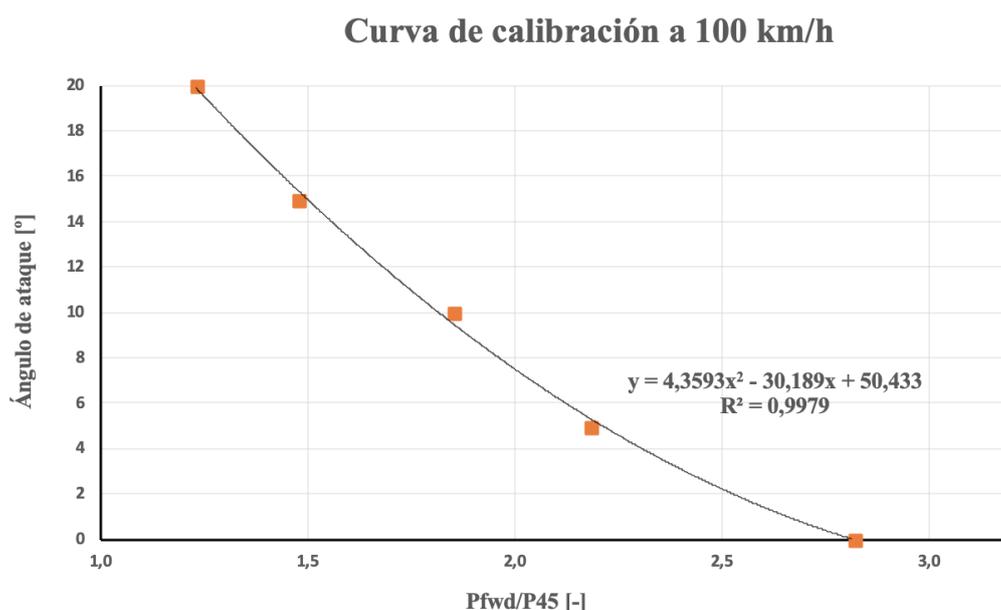


Figura 3.5: Curva de calibración para $V=100$ km/h

En ella se puede observar cómo el promedio de las medidas sigue una relación esperada, en la que, a medida que el ángulo de ataque aumenta, P_{45} aumenta (se encuentra más perpendicular al flujo) y P_{fwd} disminuye, por lo que su división, para valores del ángulo de ataque mayores, es menor.

Sin embargo, cabe destacar que la dispersión de las 90 medidas realizadas para cada uno de los ángulos es muy superior a la esperada mediante el método explicado de Rogers. Esto puede estar debido a varios factores entre los que se encuentran los siguientes:

- **Turbulencia:** El experimento llevado a cabo en la parte posterior del vehículo está afectado, casi con total seguridad, por unos niveles de turbulencia aerodinámica que no son aceptables para proceder a la calibración correcta del sistema con la precisión esperada. La sonda situada alrededor de 10 centímetros fuera del coche se vería afectada por las perturbaciones del aire incidente debidas tanto a la misma carrocería del vehículo como a la presencia, alrededor de 1.5 metros más adelante, del espejo retrovisor.
- **Rodaje del vehículo:** Si bien es cierto que se comprobó durante el experimento que las vibraciones de la tabla de madera y, por ende, de la sonda fueron mínimas respecto a la posición del vehículo teniendo en cuenta las condiciones, el rodaje del coche sobre el asfalto producía en ocasiones variaciones importantes de estos parámetros. En un experimento en el que se busca calibrar de la mejor manera unos sensores tan precisos, este factor supone una considerable fuente de error.
- **Otros factores externos:** En el túnel de viento, las condiciones externas y los valores de todos los parámetros que envuelven al experimento están controlados y son conocidos (sino todos, la gran mayoría). Sin embargo, durante el experimento llevado a cabo esto no ocurre. Entre otros factores, cabe mencionar que al estar expuesta la sonda al aire exterior, existía cierta influencia del viento incidente. En concreto, ese día había rachas de hasta 20 km/h en dirección transversal a la del viento relativo debido al desplazamiento del coche. A esto se suma la precisión de la aplicación utilizada para mantener el ángulo de inclinación constante y la dificultad de mantener el mismo a la vez que se realizaba el experimento.

Además, según lo explicado en el método de Rogers, esta curva debe ser independiente de la velocidad, por lo que debería resultar la misma para una velocidad de vuelo diferente. Sin embargo, como se ha comentado, 100 km/h ya es una velocidad menor a la de entrada en pérdida por lo que reducirla para obtener datos con otro valor no tendría interés real. Por otra parte, las limitaciones del experimento sujetas a las normas de tráfico dificultaban circular a una velocidad diferente con suficiente margen como para obtener conclusiones válidas.

Con todo, sí cabe mencionar que el resultado mostrado en la *Figura 3.5* ofrece una buena aproximación, al menos en el aspecto cualitativo, de lo que se habría esperado de la calibración en el túnel de viento bajo condiciones óptimas. Quedando como trabajo a realizar la calibración en un entorno controlado y su posterior validación, por ejemplo, volviendo a realizar las medidas con el código explicado en la sección 2.3.2 *Código* y comprobando que, efectivamente, la recta de calibración muestra por pantalla los valores de ángulo de ataque correspondientes a la inclinación controlada de la sonda, de la que se dispone el valor.

4. Conclusión

En el presente Trabajo de Fin de Grado se ha desarrollado un sistema de medida del ángulo de ataque de manera efectiva mediante el uso de dos sensores de presión diferencial.

A lo largo del mismo, se ha mostrado el funcionamiento de los dispositivos actuales que se emplean en tareas similares, dispositivos de alerta de entrada en pérdida y medidores del ángulo de ataque en aeronaves comerciales.

Se ha desarrollado el entorno electrónico y montaje general necesario para la creación del sistema, dando lugar a un prototipo útil y sencillo. Este muestra por la pantalla LCD el valor del ángulo de ataque mediante una interfaz clara y visual. Además, la implementación de la PCB ha permitido la reducción de su tamaño, facilitando así su futura introducción en la cabina de una aeronave.

En cuanto a las limitaciones, como se ha visto, no se ha podido desarrollar la parte de calibración en condiciones óptimas debido a la crisis de la Covid-19, no pudiendo, por tanto, obtener resultados en lo que a la precisión del sistema se refiere. Sin embargo, la prueba experimental realizada sí ofrece una curva de calibración con la tendencia esperada, por lo que la validación del sistema queda solo a la espera de su posible introducción en el túnel de viento.

Así pues, varias son las ampliaciones posibles de este trabajo. En primer lugar, cuando se tenga acceso, efectuar de manera controlada la calibración del prototipo y su correspondiente validación. Por otro lado, sería interesante la implementación real del sistema en una aeronave ligera, ya que el prototipo al completo está diseñado con dicho propósito desde la alimentación portátil hasta el diseño de la pieza 3D a la que se conecta el sistema.

También, en el caso de que fuera necesario si los resultados de dichos experimentos no fueran satisfactorios, se podría revisar el diseño de la pieza 3D y adaptarlo, ya que podría suponer una fuente de error. Otra solución sería la adquisición de una sonda disponible en el mercado, si bien aumentaría el coste del sistema. Se podría añadir también algún elemento más a la interfaz con el fin de que resultara aun más clara una situación de posible peligro.

Para terminar, cabe decir que el presente Trabajo de Fin de Grado ha sido de gran utilidad en varios aspectos. En primer lugar, ha servido para afianzar y profundizar en conocimientos adquiridos a lo largo del Grado en Ingeniería Aeroespacial, especialmente en el ámbito de la electrónica, informática y la tecnología aeroespacial. Además, ha permitido enfrentarse a problemas ingenieriles relacionados con el diseño y desarrollo de un prototipo físico de manera independiente, que pueda desempeñar su función de manera correcta, poniendo en práctica lo aprendido en el Grado.

5. Presupuesto

Para finalizar con este Trabajo de Fin de Grado sobre el desarrollo de un sistema de medida del ángulo de ataque para aeronaves ligeras mediante sensores de presión diferencial se especifican, a continuación, los costes aproximados para la realización del trabajo completo.

Cabe destacar que estos se han dividido en dos subapartados. El primero hace referencia al coste debido exclusivamente a la adquisición del material empleado en el desarrollo e implementación del proyecto. El segundo, por su parte, refleja los gastos derivados de la mano de obra, investigación y ensayos experimentales para el montaje y calibración del sistema. Por último se muestra el coste total del proyecto, suma de estos dos.

Coste de material

Como se ha comentado, se muestra en primer lugar el coste debido a la adquisición de los componentes necesarios para el desarrollo e implementación del proyecto.

Estos hacen referencia a la placa Arduino UNO Rev3, la pantalla, los sensores MS4525DO, el multiplexor PCA9540BD, la PCB, etc. Otros componentes tales como condensadores y resistencias se adquieren en packs de varias unidades, haciendo que su coste por unidad sea muy bajo. También cabe mencionar que la impresión de las piezas 3D de la sonda explicada en la sección *2.4 Diseño sonda 3D* no supuso coste al tener acceso directo a una impresora 3D, así como los tornillos empleados para su ensamblaje de los que ya se disponían.

Se debe tener en cuenta que la mayoría de los componentes se adquirieron a través de internet, por lo que conllevan un importe de tasas de envío. Sin embargo, en el desglose no se detallan estos costes añadidos debido a que su valor varía en gran medida. Especialmente dependiendo del tiempo del que se disponga para su obtención. Un pedido urgente puede elevar considerablemente el coste del material. Asimismo, el coste de cada uno de los componentes se puede ver afectado por el momento de desarrollo del proyecto o por una variación de precios en el mercado.

Por otro lado, al haber hecho uso de la IDE de Arduino que, como se ha visto en la sección *2.2 Arduino UNO* es de código abierto, no aparecen costes de licencias de software. Así pues, se muestra en la *Tabla 5.1* la descripción del coste de material total, con el 21 % de IVA ya aplicado sobre los componentes.

Material	Fuente	Cantidad [ud]	Coste unidad [€/ud]	Importe [€]
Arduino UNO Rev3	BricoGeek	1	24.14	24.14
Sensor MS4525DO	Digi-Key	2	36.23	72.46
Multiplexor PCA9540BD	Farnell	1	1.31	1.31
Pantalla LCD	Amazon	1	13.45	13.45
PCB	JLPCB	1	0.68	0.68
Cable USB A-B	Amazon	1	2.80	2.80
Tubo de silicona de 3 m	Rovalcaucho	2	5.45	10.90
Batería portátil 2000mAh	Amazon	1	2.50	2.50
Resistencias y condensadores	Ebay	Mult.	-	3.25
TOTAL				131.49

Tabla 5.1: Costes asociados al material necesario para el sistema

Coste de mano de obra

Para el desarrollo del Trabajo se ha empleado una cantidad de tiempo considerable destinada a la búsqueda de información, desarrollo de las ideas del proyecto sobre el funcionamiento teórico del sistema, generación del código utilizado, así como al montaje propiamente dicho del prototipo especialmente debido al soldado de los componentes.

Debido a la falta de trabajo experimental de calibración del sistema, que en un principio se iba a realizar, únicamente se incluye en el cómputo la suma de las horas dedicadas como Ingeniero Aeroespacial. Si bien no se han contabilizado de manera rigurosa las horas destinadas al desarrollo del trabajo, sí se puede realizar una aproximación bastante fiable.

El precio por hora asociado a los trabajos expuestos se ha obtenido de dividir los alrededor de 22000 que según varias fuentes puede llegar a cobrar un Ingeniero Aeroespacial recién titulado al año entre las horas de una jornada laboral de 8 horas. Esta información puede verse en la *Tabla 5.2*.

Material	Horas [h]	Coste por hora [€/h]	Importe [€]
Graduado en Ingeniería Aeroespacial	250	13	3250
TOTAL			3250

Tabla 5.2: Costes asociados a la mano de obra ingenieril

A. Fichas de datos técnicos

En este anexo se encuentran los enlaces a las fichas de datos técnicos (Datasheets) de los componentes utilizados en el desarrollo del sistema.

Microcontrolador ATmega328P de la placa Arduino UNO

Multiplexor I^2C PCA9540BD

Sensores de presión diferencial MS4525DO

Controlador ILI9341 de la pantalla gráfica LCD

B. Código calibración

Se muestra en este anexo el código en Python utilizado para el almacenaje de las variables de interés como son P_{fwd} y P_{45} en el proceso de calibración del sistema llevado a cabo en una hoja de cálculo Excel.

En primer lugar se muestra la función *serialToExcel.py* y tras esta, el script *Calibracion.py* que la llama y realiza la función de almacenar los valores.

```
1 import serial
2 import xlwt
3 from datetime import datetime
4
5 class SerialToExcel:
6
7     def __init__(self, port, speed):
8
9         self.port = port
10        self.speed = speed
11
12        self.wb = xlwt.Workbook()
13        self.ws = self.wb.add_sheet("Data from Serial", cell_overwrite_ok=
True)
14        self.ws.write(0, 0, "Data from Serial")
15        self.columns = ["Date Time"]
16        self.number = 100
17
18
19    def setColumns(self, col):
20        self.columns.extend(col)
21
22    def setRecordsNumber(self, number):
23        self.number = number
24
25    def readPort(self):
26        ser = serial.Serial(self.port, self.speed, timeout=1)
27        c = 0
28        for col in self.columns:
29            self.ws.write(1, c, col)
30            c = c + 1
31        self.fila = 2
32
33        i = 0
34        while(i < self.number):
35            line = str(ser.readline())
36            if(len(line) > 0):
37                now = datetime.now()
38                date_time = now.strftime("%y/%d/%Y, %H:%M:%S")
39                print(date_time, line)
40                if(line.find(",") > 0):
41                    c = 1
42                    self.ws.write(self.fila, 0, date_time)
```

```

43         columnas = line.split(",")
44         for col in columnas:
45             self.ws.write(self.fila, c, col)
46             c = c + 1
47
48         i = i + 1
49         self.fila = self.fila + 1
50
51     def writeFile(self, archivo):
52         self.wb.save(archivo)

```

serialToExcel.py

```

1 from serialToExcel import SerialToExcel
2
3 serialToExcel = SerialToExcel("/dev/cu.usbmodem14101",9600)
4
5 columnas = ["p45", "p45corr", "p45", "p45corr", "p45", "Alfaround"]
6
7 serialToExcel.setColumns(["p45", "p45corr", "p45", "p45corr", "p45", "Alfaround"])
8 serialToExcel.setRecordsNumber(90)
9 serialToExcel.readPort()
10
11 serialToExcel.writeFile("Calib_0_100.xls")

```

Calibracion.py

Los nombres de las variables se corresponden con las del código mostrado en la sección 2.3.2.

Bibliografía

- [1] EASA. "*Stall and Stick Pusher Training, Service Bulletin*" (2013). Accedido el 15 de mayo de 2020. URL: <https://www.skybrary.aero/bookshelf/books/2248.pdf>.
- [2] FAA. "*Upset Prevention and Recovery Training, Advisory Circular*" (2013). Accedido el 15 de mayo de 2020. URL: <https://www.skybrary.aero/bookshelf/books/3175.pdf>.
- [3] D.F. Rogers. "*Fundamental Angle of Attack*" (2013). Accedido el 11 de mayo de 2020. URL: http://www.nar-associates.com/technical-flying/angle_of_attack/FundamentalAoA_wide_screen.pdf.
- [4] Carta de la "FAA Small Airplane Directorate" a DepotStar Inc. (15 de diciembre de 2011).
- [5] D.F. Rogers. "Investigation of a General Aviation Differential Pressure Angle of Attack Probe". En: *Journal of Aircraft* 50 (2013), págs. 1668-1671.
- [6] D.F. Rogers - Borja Martos - Francisco Rodrigues. *Low-cost accurate angle of attack system*. FAA, 2018.
- [7] EASA. "*Easy Access Rules for Normal-Category Aeroplanes (CS-23)*" (última modificación Noviembre 2019). Accedido el 15 de mayo de 2020. URL: <https://www.easa.europa.eu/document-library/general-publications/easy-access-rules-normal-category-aeroplanes-cs-23>.
- [8] FAA. "*Approval of Non-Required Angle of Attack (AoA) Indicator Systems (Febrero 2014)*". Accedido el 15 de mayo de 2020. URL: [https://rgl.faa.gov/Regulatory_and_Guidance_Library/rgPolicy.nsf/0/eb0fac0c1641509586257c76005e6274/\\$FILE/AIR100-14-110-PM01.pdf](https://rgl.faa.gov/Regulatory_and_Guidance_Library/rgPolicy.nsf/0/eb0fac0c1641509586257c76005e6274/$FILE/AIR100-14-110-PM01.pdf).
- [9] *Página web del BendixKing KLR 10 'Lift Reserve Indicator'*. Accedido el 27 de mayo de 2020. URL: <https://www.bendixking.com/en/products/productitems/klr-10>.
- [10] *Página web del Garmin GI 260 AoA indicator*. Accedido el 27 de mayo de 2020. URL: <https://buy.garmin.com/es-ES/ES/p/149850>.
- [11] *Página web de EasyEDA - Simulador de circuitos y diseño de PCB*. Accedido el 27 de mayo de 2020. URL: <https://easyeda.com/es>.