



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

Implementación de Máquina de Estados Finita en  
Microcontrolador para optimizar el control del sistema de  
energía de un dron híbrido pila de combustible-batería

Trabajo Fin de Grado

Grado en Ingeniería Electrónica Industrial y Automática

AUTOR/A: Elbachouti Khalil, Walid

Tutor/a: Sánchez Díaz, Carlos

Cotutor/a: Rocha Gaso, María Isabel

Director/a Experimental: ZAZO MANZANEQUE, ROBERTO

CURSO ACADÉMICO: 2022/2023



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

TRABAJO FIN DE GRADO

INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

# Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible-batería”

**AUTOR:** Walid El Bachouti Khalil

**TUTOR:** Carlos Sánchez Díaz

**COTUTORA:** María Isabel Rocha Gaso

**DIRECTOR EXPERIMENTAL:** Roberto Zazo Manzaneque

**Curso académico:** 2022-2023



## **RESUMEN**

En tiempos modernos, se realiza un uso excesivo de combustibles fósiles en el sector de los vehículos móviles, de hecho, el transporte por carretera, junto con el transporte aéreo y el marítimo constituyen más de un 26% de las emisiones de  $CO_2$  y otros gases de efecto invernadero, según el Parlamento Europeo en 2019. Este gran porcentaje de emisiones ha obligado a la sociedad a buscar otras tecnologías para impulsar vehículos, que sean más sostenibles y respeten el medioambiente, como es la tecnología del hidrógeno y su aplicación en las pilas de combustible.

Estas pilas de combustible se encuentran en una etapa de desarrollo avanzada, pues se consideran las sustitutas de los motores de combustión interna de los vehículos actuales, ya que una de sus principales ventajas es que su único residuo tras la reacción del oxígeno y el hidrógeno es el agua.

El objetivo del presente Trabajo de Fin de Grado es el control y su posterior validación del sistema de energía de un dron o vehículo aéreo no dirigido con una doble alimentación constituida por una pila de combustible del tipo PEMFC (Proton Exchange Membrane Fuel Cell) y una batería LiPo auxiliar.

Dicho control se realiza con un programa en lenguaje C, implementando una máquina de estados finita en un microcontrolador, y pretende garantizar el funcionamiento durante la fase de vuelo estacionario mediante la pila de combustible y alcanzar los picos de demanda en los momentos de despegue y aterrizaje mediante la batería auxiliar.

Finalmente, el sistema de control implementado se validó y analizó mediante una simulación del UAV.

**Palabras clave:** hidrógeno, pilas de combustible, PEMFC, dron, batería, microcontrolador.



## **RESUM**

En temps moderns, es realitza un ús excessiu de combustibles fòssils en el sector dels vehicles mòbils, de fet, el transport per carretera, juntament amb el transport aeri i el marítim constitueixen més d'un 26% de les emissions de  $CO_2$  i altres gasos d'efecte d'hivernacle, segons el Parlament Europeu en 2019. Aquest gran percentatge d'emissions ha obligat a la societat a escalar a altres tecnologies per a l'alimentació de vehicles, que siguin més sostenibles i respecten el medi ambient, com és la tecnologia de l'hidrogen i la seua aplicació en les piles de combustible.

Aquestes piles de combustible es troben en una etapa de desenvolupament avançada, perquè es consideren les substitutes dels motors de combustió interna dels vehicles actuals, ja que una dels seus principals avantatges és que el seu únic residu després de la reacció de l'oxigen i l'hidrogen és l'aigua.

L'objectiu del present Treball de Fi de Grau és el control optimitzat d'un dron o vehicle aeri no dirigit amb una doble alimentació constituïda per una pila de combustible del tipus PEMFC i una bateria LiPo auxiliar.

Aquest control es realitza amb un programa en llenguatge C, implementant una màquina d'estats finita en un microcontrolador, i pretén garantir el funcionament durant la fase de vol estacionari mitjançant la pila de combustible i aconseguir els pics de demanda en els moments d'enlairament i aterratge mitjançant la bateria auxiliar.

Finalment, el sistema de control implementat es validarà i analitzarà mitjançant una simulació del UAV.

**Paraules clau:** hidrogen, piles de combustible, PEMFC, dron, bateria, microcontrolador.



## **ABSTRACT**

Today, there is an excessive use of fossil fuels in the vehicle sector, in fact, road transport, together with air and sea transport constitute more than 26% of the emissions of  $CO_2$  and other greenhouse gases, according to the European Parliament in 2019. This large percentage of emissions has forced society to look for other more sustainable and environmentally friendly technologies for powering vehicles, such as hydrogen technology and its application in fuel cells.

These fuel cells are at an advanced stage of development, as they are considered to replace the internal combustion engines of current vehicles, since one of their main advantages is that their only residue after the reaction of oxygen and hydrogen is water.

The aim of this Final Degree Project is the optimised control of a drone or unmanned aerial vehicle with a dual power supply consisting of a PEMFC fuel cell and an auxiliary LiPo battery.

This control is carried out with a programme in C language, which implement a finite state machine in a microcontroller, and aims to guarantee the operation during the stationary flight phase with the fuel cell and to reach peak demand during take-off and landing by the use of the auxiliary battery.

Finally, the implemented control system will be validated and analysed by means of a UAV simulation.

**Keywords:** hydrogen, fuel cells, PEMFC, drone, battery, microcontroller.





## **ÍNDICE GENERAL DEL TRABAJO FIN DE GRADO**

- Memoria
- Planos
- Pliego de condiciones
- Presupuesto
- Anexo I
- Anexo II
- Anexo III
- Anexo IV







UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



# MEMORIA

**Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”**



## **ÍNDICE DE LA MEMORIA**

<b>1. MOTIVACIÓN, JUSTIFICACIÓN Y OBJETIVOS .....</b>	<b>1</b>
1.1 Justificación académica .....	2
1.2 Justificación profesional .....	2
1.3 Objetivos .....	3
<b>2. INTRODUCCIÓN .....</b>	<b>5</b>
2.1 Vehículos aéreos no tripulados: Drones .....	6
2.2 El hidrógeno como vector energético .....	10
2.3 Pilas de combustible .....	12
2.4 Baterías de Litio Polímero .....	15
2.5 Microcontroladores.....	19
<b>3. DESCRIPCIÓN DE LA SOLUCIÓN.....</b>	<b>24</b>
3.2 Arquitectura del sistema .....	24
3.3 Diseño del hardware .....	32
3.3.1 Batería .....	35
3.3.2 Sensor de corriente .....	38
3.3.3 UART .....	40
3.3.4 Filtro Paso Bajo (FPB) .....	43
3.4 Diseño del software .....	44
3.4.1 Lógica de Control .....	45
3.4.2 El control .....	47
<b>4. RESULTADOS Y DISCUSIÓN .....</b>	<b>58</b>
<b>5. BIBLIOGRAFÍA .....</b>	<b>64</b>

## **ÍNDICE DE FIGURAS**

Figura 1. Esquema general del sistema híbrido eléctrico “batería – pila de combustible” como fuente de energía para un dron o UAV de gran tamaño [1]. .....	4
Figura 2. Dron con batería LiPo modelo Mavic Mini, de la empresa DJI [4]. .....	7
Figura 3. Dron híbrido con "batería - mecánica de combustión interna", modelo HIBRiX.20, marca Quaternium (España) [6]. .....	8
Figura 4. Dron híbrido con batería y pila de combustible de la marca H2DronEnergy [7]. .....	8
Figura 5. Esquema representativo de la economía circular del hidrógeno [8]. .....	12
Figura 6. Esquema conceptual del funcionamiento de una pila de combustible de tipo PEMFC [12]. .....	14
Figura 7. Funcionamiento general de las baterías LiPo [13]. .....	16
Figura 8. Esquema del funcionamiento de las baterías de Litio Polímero en descarga. ....	17
Figura 9. Esquema de la arquitectura Von Neumann. ....	19
Figura 10. Esquema de la arquitectura Harvard. ....	20
Figura 11. Esquema del conjunto del microcontrolador. ....	22
Figura 12. Esquema representativo del sistema de alimentación "pila de combustible - batería" de un dron o vehículo aéreo no tripulado. ....	25
Figura 13. Batería LiPo 4S1P empleada (14.8 V de tensión nominal y 3300 mA/h). ....	28
Figura 14. Carga electrónica empleada para simular el comportamiento del dron. ....	28
Figura 15. Esquema representativo del microcontrolador conectado al ordenador. ....	29
Figura 16. Placa de desarrollo STM32F429I-DISC1 con microcontrolador STM32F429ZIT6 [21].	29
Figura 17. Interfaz de usuario para configurar los pines del microcontrolador STM32F429ZIT6 en la plataforma STM32CubeIDE. ....	30
Figura 18. Parte superior (top) de la tarjeta de interfaz de hardware diseñada. ....	33
Figura 19. Parte inferior (bottom) de la tarjeta de interfaz de hardware diseñada. ....	33
Figura 20. Pila de combustible, modelo H100 de la marca h2planet [23]. ....	34
Figura 21. Regulador lineal L7805C. ....	37
Figura 22. Sensor de corriente ACS712 integrado en placa de circuito impreso [24]. ....	38
Figura 23. Esquema conceptual de la transmisión y recepción de datos desde el microcontrolador al PC y viceversa. ....	41
Figura 24. Adaptador de DB-9 a USB para la conexión con el PC. ....	43
Figura 25. Diagramas de Estado de Moore y de Mealy. ....	46
Figura 26. Configuración de los periféricos. ....	53



# Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

Figura 27. Estados, transiciones y eventos de las FSM en C.....	53
Figura 28. Disparo de las FSM.....	54
Figura 29. Declaración de las funciones de guarda y acción.....	54
Figura 30. Conversión de datos tras la lectura del ADC2.....	55
Figura 31. Ejemplos funciones de guarda.....	55
Figura 32. Ejemplos funciones de acción.....	56
Figura 33. Ejemplo función de acción sobre el DAC2.....	56
Figura 34. Ejemplo de definiciones de constantes en el main.h.....	57
Figura 35. Esquema de las conexiones para la validación del sistema.....	58
Figura 36. Validación de lectura de tensiones de batería.....	59
Figura 37. Lectura de las tensiones de celda de la batería.....	59
Figura 38. Validación lectura de corriente con la carga electrónica y el sensor de corriente.....	60
Figura 39. Conexionado del sistema completo.....	60
Figura 40. Medición con el osciloscopio de uno de los canales ADC.....	61
Figura 41. Validación del relé de seguridad y la válvula principal en el modo depuración.....	62
Figura 42. Validación del funcionamiento de los canales DAC y la válvula de purgado.....	63

## **ÍNDICE DE TABLAS**

Tabla 1. <i>Comparativa de 3 modelos de drones con fuentes de energía de baterías, motores de combustión y pilas de combustible.</i> .....	9
Tabla 2. <i>Tipos de hidrógeno según su método de producción y sus emisiones.</i> .....	11
Tabla 3. <i>Comparación de la temperatura de operación (°C), la potencia y la eficiencia de las principales Pilas de Combustible.</i> .....	13
Tabla 4. <i>Tabla comparativa de las baterías LiPo, Li-ion y ácido-plomo.</i> .....	18
Tabla 5. <i>Clasificación y descripción de las variables del sistema "batería-pila de combustible" del dron.</i> .....	31
Tabla 6. <i>Valores de los condensadores a conectar al MAX232 según la tensión de alimentación.</i> .....	42



## **1. MOTIVACIÓN, JUSTIFICACIÓN Y OBJETIVOS**

Para un futuro Ingeniero Electrónico Industrial y Automático titulado en la Universidad Politécnica de Valencia (UPV), se ha convertido en una nueva necesidad del mundo tecnológico el avance de los sistemas de vuelo no tripulados, el estudio e implementación de nuevas fuentes de energía renovables que permitan mayores tiempos de vuelo y la correcta programación de microcontroladores ( $\mu\text{C}$ ) para automatizar y optimizar procesos de estos tipos de vehículos.

La mayoría de los vehículos que están basados en el uso exclusivo de baterías, tienen las limitaciones de: escasez de puntos de recarga, largos tiempos de recarga y poca autonomía. Particularmente, en el caso de los vehículos aéreos no tripulados, una limitación adicional radica en el gran peso que supone usar una batería de alta capacidad.

El avance hacia nuevas fuentes de alimentación alternativas, como es la tecnología de las pilas de combustible, permite generar energía eléctrica para dichos vehículos no tripulados u otros tipos de vehículos y reducir drásticamente la emisión de gases de efecto invernadero mediante el elemento más abundante del planeta, el hidrógeno ( $H_2$ ).

El uso de los vehículos aéreos no tripulados (conocidos comúnmente como drones), se ha extendido en diversas áreas, y son de mucha utilidad en sectores como en la topografía, en inspecciones técnicas, en el sector agrario, o en emergencias y seguridad. Por este motivo, ya que los drones son muy eficaces en labores de rescate de personas, vigilancia o detección de incendios y su monitorización, es muy importante la implementación de nuevas tecnologías de alimentación eléctrica para extender la autonomía y reducir los tiempos de recarga.

Por otra parte, se ha extendido también el uso de microcontroladores para automatizar procesos y adquirir datos, ya que, permiten, entre otras tareas, leer entradas en tiempo real, con el uso de dispositivos como los convertidores de analógico a digital (ADC) y controlar salidas como pueden ser electroválvulas con los convertidores de digital a analógico (DAC).

Teniendo en cuenta todas estas consideraciones, el presente Trabajo de Final de Grado (TFG), pretende realizar un control optimizado del sistema de alimentación de un dron híbrido con una batería LiPo que contribuirá en los picos altos de demanda, y una pila de combustible que utiliza el hidrógeno para generar energía. Dicho control se hace mediante la correcta programación del microcontrolador. En el TFG, se realiza una validación del sistema con una carga electrónica que actúa como dron para validar el control implementado.

## **1.1 Justificación académica**

El presente TFG pretende confirmar que el alumno ha adquirido los conocimientos y las competencias requeridas en su titulación. Además, servirá para la obtención de los 12 créditos ECTS necesarios para alcanzar el título de Ingeniería Electrónica Industrial y Automática por la Universidad Politécnica de Valencia (UPV).

## **1.2 Justificación profesional**

Desde una perspectiva profesional, este avance hacia nuevas fuentes de alimentación utilizando el hidrógeno como combustible para vehículos aéreos, terrestres u otros transportes suponen, por una parte, una reducción drástica de la emisión de gases de efecto invernadero, que son una de las principales causas que contribuyen al calentamiento global, y, por otra parte, un avance en el autoabastecimiento energético.

Además, el uso de microcontroladores para controlar el funcionamiento de una tarea determinada, como es el caso de las corrientes que fluyen por el sistema de alimentación con pila de combustible o por la batería del dron, se ha extendido debido a su velocidad de procesamiento y su pequeño tamaño, entre otras ventajas, que permite su integración en el propio dispositivo al que gobierna.

Los resultados del presente TFG pueden suponer un impulso hacia la incorporación de estos sistemas en otros tipos de vehículos, consiguiendo así solventar unos de los grandes problemas de la sociedad actual como son la escasez de combustibles fósiles y la emisión de gases de efecto invernadero, a la vez de mejorar la autonomía de los sistemas que utilizan únicamente baterías.

### 1.3 Objetivos

El objetivo del presente TFG es el control optimizado mediante la implementación de una máquina de estados finita en un microcontrolador del sistema de energía híbrido de un vehículo aéreo no tripulado (*Unmanned Aerial Vehicles, UAV*), conocidos comúnmente como drones. Dicho sistema de energía está compuesto por una batería y una pila de combustible, que utiliza el hidrógeno como vector energético.

En la actualidad, la gran mayoría de los drones comerciales utilizan como fuente de energía únicamente baterías de Litio Polímero (*LiPo*), las cuales se caracterizan por ser ligeras y poder almacenar una gran cantidad de energía. Sin embargo, aunque tienen un reducido tamaño y peso, en este tipo de baterías se producen descargas de energía muy elevadas para poder abastecer los picos altos de demanda de corriente eléctrica, como es el caso de los momentos de despegue o aterrizaje, lo que reduce drásticamente la autonomía de vuelo del dron y obliga a realizar cambios de batería regularmente.

Es por ello por lo que se ha incorporado una pila de combustible con un pequeño depósito de hidrógeno. Este sistema de generación de energía eléctrica permite aumentar la autonomía de vuelo, pudiendo así recorrer mayores distancias y reducir los largos tiempos de recarga de baterías.

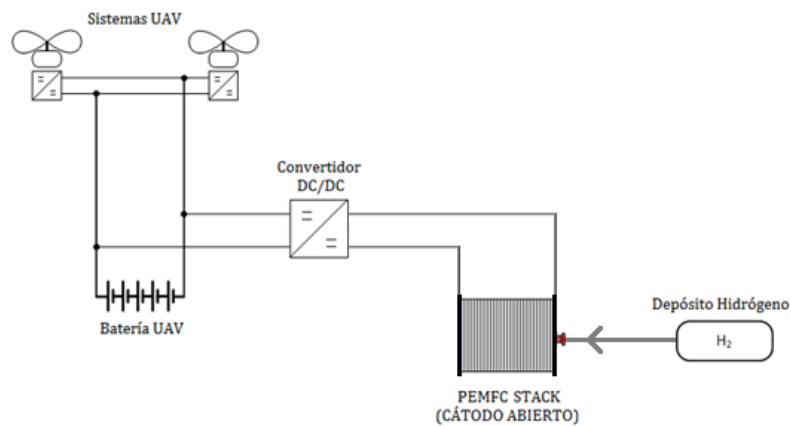
Sin embargo, dicha incorporación supone un gran aumento del peso total del dron, llegando a alcanzar pesos superiores a los 20 kg, por lo que un dron de estas características será adecuado para aplicaciones que operen en situaciones de emergencia, rescate, vigilancia o grabación de imágenes aéreas [1].

El uso de una batería en el sistema híbrido eléctrico es necesario para cubrir los picos altos de demanda de corriente en las fases de encendido y apagado de los componentes de arranque y parada del dron. Se ha optado por una batería LiPo ya que ofrecen una gran densidad energética en combinación con un peso y tamaño reducido.

En la siguiente Figura 1, se muestra un esquema general que representa el sistema de alimentación híbrido eléctrico “batería – pila de combustible” para un dron o UAV de gran tamaño:

# Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---



**Figura 1.** Esquema general del sistema híbrido eléctrico “batería – pila de combustible” como fuente de energía para un dron o UAV de gran tamaño [1].

Para acondicionar la tensión que proporciona la pila de combustible a la batería, se utiliza un convertidor DC/DC. Por lo tanto, es de gran importancia realizar el control optimizado de la corriente eléctrica producida por la pila de combustible y la etapa de potencia, que alimentan el dron y cumplen la función de recarga de la batería en las fases de vuelo estacionario.

En el software diseñado se ha implementado una máquina de estados finitos para el control de todas las variables, la cual ayuda a entender mejor el código y a reducir la cantidad de variables globales necesarias.

## **2. INTRODUCCIÓN**

Con la quema de combustibles fósiles para el transporte se producen emisiones de grandes cantidades de Dióxido de Carbono ( $CO_2$ ), que es uno de los gases responsables del incremento del denominado efecto invernadero y del aumento de la temperatura media del planeta. De hecho, el  $CO_2$  es la sustancia que más está contribuyendo a este efecto, absorbiendo gran parte de la radiación solar, y reteniéndola próxima a la superficie del planeta [2].

Por ello, es cada vez más importante buscar nuevas fuentes de energías renovables, que son fuentes limpias, inagotables y crecientemente competitivas. Estas fuentes de energía se distinguen de los combustibles fósiles por no producir emisiones de gases de efecto invernadero ni emisiones contaminantes, y por sus costes, los cuales tienden a bajar a lo largo del tiempo.

En la actualidad, la tecnología más extendida y con más popularidad en el sector del transporte es la de los motores eléctricos, los cuales no emiten gases de efecto invernadero y son sostenibles con el medioambiente, pero cuenta con algunos inconvenientes como el peso de la batería, la poca autonomía que ofrecen en comparación con los motores de combustión interna, o la escasez de puntos de recarga. Esta tecnología es también la más extendida en los vehículos aéreos no tripulados o drones, donde las baterías, que suelen ser de Litio Polímero (LiPo) suponen un gran problema por su peso, dimensiones y en especial, por la corta duración de los vuelos debido a su poca capacidad.

Como solución a la escasa autonomía que ofrecen las baterías, la tecnología de las pilas de combustible se muestra como una posible alternativa a los combustibles fósiles para la alimentación de vehículos, ya que es capaz de generar energía eléctrica usando el hidrógeno y oxígeno que, produce agua y calor como residuos.



Por otro lado, el uso de microcontroladores y microprocesadores en la industria está siendo cada vez más importante, ya que ofrecen la posibilidad de realizar múltiples tareas, y en concreto, en el sector automotriz estos equipos se utilizan para procesar los datos que provienen de los diferentes sensores del vehículo, controles de transmisión o sistemas de frenado, entre otros usos [3].

Los microcontroladores permiten programar y controlar sistemas de manera óptima, y normalmente cuentan con una CPU (Unidad Central de Procesamiento), buses, memoria, varias entradas y salidas controlables, y periféricos como los temporizadores, los que convierten señales analógicas a digitales (ADC, *Analog to Digital Converter*) o los que convierten señales digitales a analógicas (DAC, *Digital to Analog Converter*).

Los microprocesadores son elementos que únicamente realizan operaciones lógicas aritméticas, y, aunque estas operaciones las realizan más rápidamente en estos dispositivos, necesitan de más periféricos adicionales para su funcionamiento, como los controladores de buses o memorias, que acaba haciendo que el dispositivo aumente bastante de tamaño y de precio.

## **2.1 Vehículos aéreos no tripulados: Drones**

Los drones (UAV) son máquinas que vuelan sin que se encuentre físicamente una persona en el aparato. Son dirigidas remotamente por telemando en tiempo real o por un programa informático para que siga unas instrucciones y parámetros específicos.

Los drones han transformado varios ámbitos de aplicación, ya que son capaces de realizar funciones civiles, como vigilar el tráfico o cultivos agrícolas, retransmitir imágenes aéreas, entregar mercancías, transportarlas, realizar servicios forestales, de emergencia, de rescate y de salvamento de personas, entre muchas otras aplicaciones.

A continuación se hará una revisión de los drones más relevantes para el trabajo disponibles comercialmente en la actualidad.

En la Figura 2, se muestra el dron *Mavic Mini* de la empresa *DJI*, un vehículo aéreo no tripulado que utiliza una batería de Litio Polímero (LiPo), el cual destaca por su pequeño tamaño y peso, que no supera los 249 gramos:



**Figura 2.** Dron con batería LiPo modelo Mavic Mini, de la empresa DJI [4].

El modelo *Mavic Mini* que se muestra en la Figura 2 tiene 3 baterías, las cuales ofrecen una autonomía de 25 minutos cada una, dependiendo del estilo de vuelo y las condiciones meteorológicas del momento.

También existen drones que funcionan con una tecnología híbrida de “batería – mecánica de combustión interna”, u otros con tecnología híbrida de “batería- pila de combustible de hidrógeno” los cuales incrementan notablemente la autonomía, pero que aún cuentan con algunos problemas, como la necesidad de diseñar un dron de grandes dimensiones para poder incorporar la mecánica de combustión interna y el depósito de gasolina, o la pila de combustible y el depósito de hidrógeno, por lo que aumenta también el peso y, por consecuencia, el consumo.

Por lo tanto, este tipo de drones con tecnologías híbridas no deben ser utilizados en labores de entrega de paquetería o en vuelos de alta velocidad, sino que son más aptos para labores de rescate o vigilancia, ya que combinan la capacidad de soportar pesos externos y la larga duración de vuelo, que permite hacer extensas vigilancias [5].

Por una parte, en la Figura 3, se muestra un dron híbrido con “batería – mecánica de combustión interna”, modelo HIBRiX.20 de la marca *Quaternium*, el cual tiene incorporado un depósito de gasolina de 10 Litros, que le permite un vuelo de más de 4 horas:



**Figura 3.** Dron híbrido con "batería - mecánica de combustión interna", modelo HIBRiX.20, marca Quaternium (España) [6].

Este modelo de dron, al funcionar con un motor de gasolina, es un dron contaminante, y tiene otras desventajas relacionadas con los motores de combustión, como el ruido o el peso, pero ganan sus ventajas.

Entre las principales ventajas destacan la gran autonomía y rendimiento que ofrecen, ya que son capaces de realizar grandes recorridos a grandes velocidades sin parar a repostar y evitando largos tiempo de recarga de la batería.

Por otra parte, en la Figura 4 se observa un dron híbrido con “batería – pila de combustible de hidrógeno”, el cual tiene un gran alcance y una gran autonomía de vuelo, y dispone de un sistema de baterías que se recargan en el vuelo y permiten alcanzar los picos de potencia en las fases de vuelo más exigentes:



**Figura 4.** Dron híbrido con batería y pila de combustible de la marca H2DronEnergy [7].

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

Este tipo de drones utilizan una pila de combustible para alimentar el vehículo y recargar el sistema de baterías, que son importantes para alcanzar los picos de demanda de consumo en las fases de vuelo más demandantes, como el despegue o el aterrizaje, y son una gran opción para sustituir a los drones convencionales que funcionan únicamente con baterías, ya que, al igual que estos drones convencionales, no emiten gases que agudicen el efecto invernadero ni al calentamiento global, pero ofrecen más autonomía y menos tiempos de recarga.

En la Tabla 1, se muestra una pequeña tabla resumen comparando los 3 modelos de drones mencionados anteriormente:

**Tabla 1.** Comparativa de 3 modelos de drones con fuentes de energía de baterías, motores de combustión y pilas de combustible.

<b>Modelo del dron</b>	<i>Mavic Mini</i>	<i>HIBRiX.20</i>	<i>H2DronEnergy</i>
<b>Fuente de energía</b>	Únicamente batería	Batería y motor de combustión interna (gasolina)	Batería y pila de combustible (hidrógeno)
<b>Autonomía</b>	Hasta 30 min	Hasta 4 h (240 min)	Hasta 3 h (180 min)
<b>Peso</b>	0.249 kg	Hasta 25 kg	Hasta 25 kg
<b>Aplicaciones</b>	Uso recreativo, carreras, fotografía, inspección de techos.	Agricultura, cartografía, inspección industrial, seguridad, etc.	Agricultura, vigilancia y seguridad, minería, fotografía.

Se observa que la tecnología del motor de combustión sigue siendo la que más autonomía tiene, pero con los nuevos avances sobre las pilas de combustible se han conseguido autonomías de vuelo muy parecidas.

Las aplicaciones de los diferentes drones dependen sobre todo de la autonomía y el peso que este tiene, por lo que, un dron con mayor peso será más adecuado para actividades que requieran un largo tiempo sin recargar, o que necesiten cargar materiales o tener una cámara, y los drones con menor peso serán más adecuados para actividades recreativas, como carreras, o inspecciones en lugares donde los grandes drones no pueden alcanzar a llegar, etc.

## **2.2 El hidrógeno como vector energético**

El hidrógeno ( $H$ ) es un elemento normalmente gaseoso, el más abundante del universo, y es el combustible de las estrellas, incluida la nuestra, el Sol. Es un vector energético y no contaminante, del que se puede obtener energía mecánica, térmica o eléctrica, y ser posteriormente liberada de forma controlada.

La posibilidad de su almacenamiento y su energía específica de combustión han hecho que este elemento sea importante, aunque, para conseguir hidrógeno en su forma diatómica ( $H_2$ ) se deben realizar procesos de transformación mediante la electrólisis del agua [1]. Este elemento, en su forma diatómica es usado en medios de transporte, donde se produce agua ( $H_2O$ ) como único residuo tras una combustión limpia.

Una de las mayores ventajas del uso del hidrógeno es la posibilidad de generar energía eléctrica. Esto se consigue canalizando el hidrógeno en su forma diatómica procedente de un tanque específico hacia una pila de combustible, que es donde se produce la reacción con el oxígeno ( $O_2$ ) procedente del aire y se consigue energía eléctrica.

Para obtener el hidrógeno existen diversos métodos cuyo principio de funcionamiento es separar el hidrógeno de otros elementos químicos, como del carbono en los combustibles fósiles o del oxígeno en el caso del agua.

Dependiendo del método de producción aplicado, se pueden distinguir distintos tipos de hidrógeno, como el verde, azul, gris, turquesa, marrón o rosa.

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

En la Tabla 2 se muestran los distintos tipos de hidrógeno, su método de producción, y las emisiones asociadas a ese proceso:

**Tabla 2.** *Tipos de hidrógeno según su método de producción y sus emisiones.*

<b>Hidrógeno</b>	<b>Método de producción</b>	<b>Emisiones</b>
<b>Verde</b>	Electrólisis a través de energías renovables	Muy bajas
<b>Azul</b>	Extracción a partir de hidrocarburos con captura de emisiones contaminantes	Medias
<b>Gris</b>	Reformado por vapor a partir de gas natural sin captura de emisiones contaminantes	Altas
<b>Turquesa</b>	Pirólisis de metano	Bajas
<b>Marrón</b>	Gasificación del carbón	Muy altas
<b>Rosa</b>	Electrólisis a partir de energía nuclear	Bajas

Cabe destacar de entre todos los tipos el hidrógeno verde, el cual es la forma más ecológica de producir hidrógeno, con emisiones de Dióxido de Carbono (CO<sub>2</sub>) prácticamente nulas.

Actualmente, la mayor parte del hidrógeno que se produce se hace a partir de hidrocarburos, que producen emisiones contaminantes, de hecho, el hidrógeno más utilizado por la industria es el azul, ya que su coste es cada vez más competitivo.

La producción competitiva del hidrógeno verde y su uso como vector energético podría ser la clave en el camino hacia la descontaminación del planeta, por eso existe la propuesta de un modelo económico energético circular para la generación, gestión y su uso como vector energético bajo en emisiones, sustituyendo a los combustibles fósiles. En la siguiente Figura 5, se muestra un esquema representativo de la economía del hidrógeno:

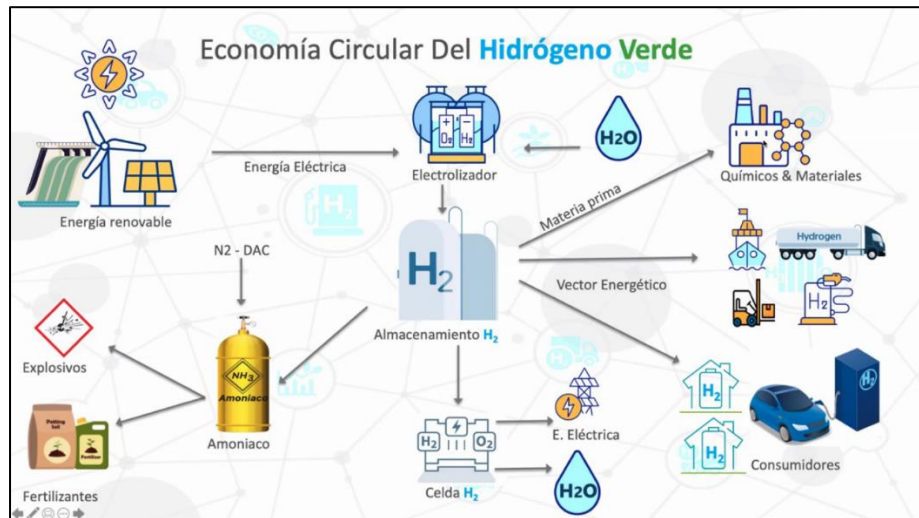


Figura 5. Esquema representativo de la economía circular del hidrógeno [8].

Se observa, por un lado, que todo el hidrógeno se genera a partir del proceso de electrólisis, y que esta propuesta permite ajustar más la producción de energía frente al consumo, ya que, al realizar la combustión del hidrógeno en las pilas de combustible, se produce agua como residuo, que puede volver a utilizarse para producir hidrógeno diatómico mediante electrólisis del agua. Dicha electrólisis del agua se propone realizarla mediante fuentes de energía primarias de origen renovable, creando así una economía circular y reduciendo drásticamente la emisión de gases de efecto invernadero.

La Asociación Española del Hidrógeno comunicó que, con la implementación de estas tecnologías, en España se evitaría la emisión de 15,2 millones de toneladas de Dióxido de Carbono ( $CO_2$ ) al año [9].

## 2.3 Pilas de combustible

La manera de generar energía eléctrica directamente en forma de corriente continua a partir de energía química (proceso electroquímico) es mediante una pila de combustible, que es un dispositivo que necesita un combustible, generalmente hidrógeno en su forma diatómica ( $H_2$ ) y un carburante u oxidante, normalmente el oxígeno ( $O_2$ ). Estas pilas constan de dos electrodos, un electrolito y las placas bipolares (ver Figura 6).

Los electrodos son normalmente dos, por una parte, el ánodo, que es donde la molécula de  $H_2$  (hidrógeno diatómico) se reduce en dos protones  $H^+$ , y por otra parte el cátodo, que es donde los protones  $H^+$  y  $O_2$  reaccionan. El electrolito se encarga de separar los gases, permitir el paso de iones  $H^+$  al cátodo y separar los electrones ( $e^-$ ). Debido al paso del flujo de electrones a través de un conductor que conecta ambos electrodos (de ánodo a cátodo), se genera energía eléctrica en forma de corriente continua. Por último, las placas bipolares, dirigen los gases y evacúan el agua ( $H_2O$ ) que se produce como residuo [10].

Entre los tipos de pilas de combustible que se encuentran en estado de comercialización o investigación destacan las Pilas de Membrana de Intercambio Protónico (PEM), las Pilas de Metanol (DMFC), las Celdas de Combustible Alcalinas (AFC), las Pilas de Combustible de Ácido Fosfórico (PAFC), las Pilas de Combustible de Carbonato fundido (MCFC) y las Pilas de Combustible de Óxido Sólido (SOFC).

En la siguiente Tabla 2, se muestra una pequeña comparativa entre los tipos de pila que se acaban de mencionar, de algunos de los factores más importantes a la hora de elegir un tipo de pila, como son la temperatura de operación, la potencia y la eficiencia eléctrica:

**Tabla 3.** Comparación de la temperatura de operación ( $^{\circ}C$ ), la potencia y la eficiencia de las principales Pilas de Combustible.

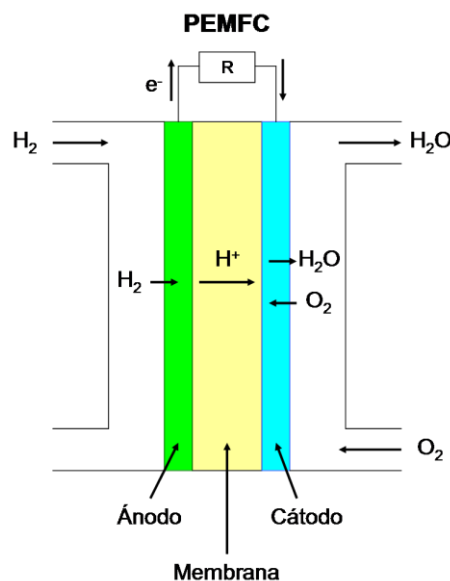
Tipo de Pila	Temperatura de operación ( $^{\circ}C$ )	Potencia	Eficiencia Eléctrica (%)
PEM	30-100	0.1 kW – 100 kW	40 - 60
DMFC	50-120	1 mW – 100 kW	20 – 30
AFC	105-245	10 kW – 100 kW	60 – 70
PAFC	180-205	>10 MW	55
MCFC	650	100 W	50-60
SOFC	800-1000	>100 kW	60-65



También es posible basarse en el material del catalizador o del electrolito, en la velocidad de arranque, su durabilidad, su coste o en sus principales aplicaciones.

Normalmente, este tipo de pilas se comercializan en forma de “*Stacks*”, que son varias celdas individuales apiladas y conectadas eléctricamente en serie. Esta técnica de apilar varias celdas individuales aumenta el rendimiento y capacidad de la pila de combustible. También pueden comercializarse con componentes como sistemas de suministro de combustible, depósitos de combustible o sistemas de enfriamiento.

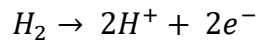
La pila de combustible de tipo PEMFC (*Proton-Exchange Membrane Fuel Cell* en inglés) o pila de combustible de membrana de intercambio protónico, es la pila de combustible más empleada en el sector de los vehículos de transporte por su alta eficiencia eléctrica (entre el 40% y el 60%), su baja temperatura de operación (entre 80 y 95 °C) y su arranque rápido [11]. En la siguiente Figura 6 se muestra el esquema conceptual del funcionamiento de dichas pilas tipo PEMFC:



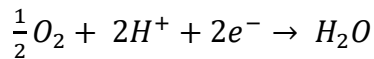
**Figura 6.** Esquema conceptual del funcionamiento de una pila de combustible de tipo PEMFC [12].

Las reacciones químicas producidas en la pila de combustible de la Figura 6 son las siguientes:

- En el ánodo, donde la molécula de hidrógeno se descompone y libera electrones (oxidación):



- En el cátodo, donde el oxígeno reacciona con los electrones liberados en la reacción anterior (reducción):



- Reacción global:  $H_2 + \frac{1}{2}O_2 \rightarrow H_2O$

Una de las consideraciones para tener en cuenta en las pilas tipo PEM es la gestión del agua, ya que en este tipo de pilas la membrana debe mantenerse hidratada, y requiere evaporar el agua en la misma cantidad en que se produce. Por un lado, si el agua se evapora más rápidamente, la membrana se seca y por consecuencia la resistencia interna aumenta. Por otro lado, si se evapora más lentamente, los electrodos se inundan [11].

Debido al pequeño diferencial de potencial generado en la pila de combustible (aproximadamente 0.7 V), lo más común es que se realicen combinaciones en serie mediante materiales con alta conductividad eléctrica para aumentar el diferencial de potencial según la aplicación deseada. Además, es normalmente una necesidad el diseño de un sistema acondicionador de la señal de salida de la pila para obtener con precisión el voltaje requerido.

## 2.4 Baterías de Litio Polímero

El principio de funcionamiento de las baterías de Litio Polímero (abreviadamente LiPo) es el intercambio de electrones ( $e^-$ ) entre el material del electrodo negativo (también llamado cátodo) y el material del electrodo positivo (también llamado ánodo). Como se observa en la Figura 7, entre ambos electrodos se encuentra otro material que cuenta con poros que permiten el paso únicamente de los iones, y evitan que dichos electrodos tengan contacto directo.

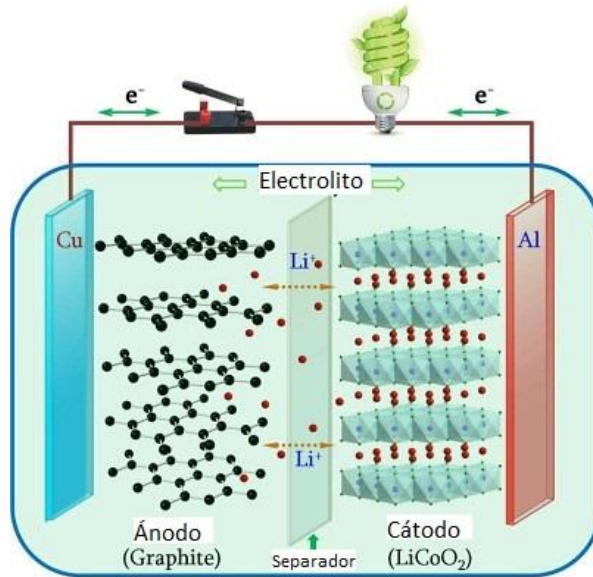
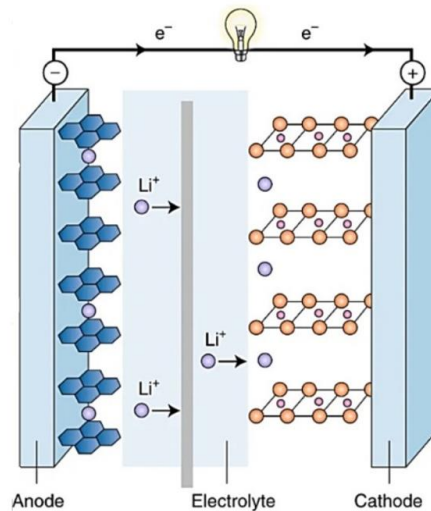


Figura 7. Funcionamiento general de las baterías LiPo [13].

Los materiales de los electrodos de estas baterías son sobre todo metales pesados, como por ejemplo el cobalto (Co), Manganeseo (Mn) o Níquel (Ni), ya que cuentan con una alta densidad energética y una vida útil bastante larga [14].

En las baterías de iones de Litio Polímero dichos iones se desplazan en un sentido cuando la batería está en fase de carga, es decir, cuando está absorbiendo la energía que le está proporcionando una fuente externa, como un cargador de baterías conectado a la red, y se desplazan en el otro sentido cuando está en fase de descarga, es decir, cuando la batería está suministrando energía a otro dispositivo.

En la Figura 8 se observa un esquema del funcionamiento del proceso de descarga de una batería de iones de Litio Polímero, donde los iones de Litio, que están representados con puntos de color amarillo, se desplazan desde el ánodo hacia el cátodo. Al mismo tiempo, los electrones ( $e^-$ ) también se desplazan desde el ánodo al cátodo, pero a través de un circuito exterior, como vemos en la Figura 8, generando una corriente eléctrica capaz de alimentar otros dispositivos [15]:



**Figura 8.** Esquema del funcionamiento de las baterías de Litio Polímero en descarga.

Al contrario que la Figura 7, cuando la batería está en fase de carga, que es cuando se conecta una fuente de energía externa (la red eléctrica normalmente) a la batería, los iones de Litio se desplazan desde el cátodo hacia el ánodo, a través del electrolito, al igual que los electrones, que lo hacen por el circuito externo. Los iones de Litio que llegan al ánodo se combinan con los electrones que también llegan, quedando el Litio depositado en dicho ánodo como consecuencia.

Algunas de las ventajas que presentan este tipo de baterías son su gran capacidad y su reducido peso y volumen. Normalmente este tipo de baterías están formadas por varias celdas, teniendo cada una un voltaje nominal de 3.7 V (voltaje de célula nominal) y un voltaje máximo de 4.2 V, todo dependiendo de los materiales empleados. Estas celdas son idénticas y se conectan en paralelo, ya que así aumenta la capacidad de la corriente de descarga y el voltaje total.

Existen otros tipos de baterías, como las baterías de ácido-plomo, que se están implementando en automóviles y motocicletas, como en los motores de arranque. Estas baterías tienen capacidad de recarga, y cuentan con dos electrodos de plomo [16]. Pero, las baterías de ácido-plomo, en comparación con las baterías LiPo, tienen un mayor peso y volumen, sus ciclos de vida son más limitados y tienen una menor densidad de energía.

Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

Otro tipo de baterías comúnmente utilizado en automóviles o motocicletas son las baterías de iones de Litio (Li-ion), que, por un lado, tienen algunas ventajas sobre las LiPo, como una mayor vida útil y una mayor seguridad, pero, por otro lado, tienen una menor densidad energética (es decir, menor peso y volumen con la misma capacidad) y una menor tasa de descarga [17].

**Tabla 4.** *Tabla comparativa de las baterías LiPo, Li-ion y ácido-plomo.*

Batería	LiPo	Ácido - plomo	Li - ion
<b>Mantenimiento</b>	Revisión básica	Limpieza de bornes, control de carga, carga mensual, añadido de agua 4 veces al año	Revisión básica
<b>Durabilidad</b>	Mas de 1000 ciclos	De 500 a 800 ciclos	De 400 a 1200 ciclos
<b>Eficiencia de carga/descarga</b>	99.8%	50-95%	80-90%
<b>Velocidad de autodescarga</b>	1%	3-20%	1%
<b>Voltaje proporcionado</b>	Voltaje nominal de 3.7 V	De 2.10 a 2.13 V	Voltaje nominal entre 3.6 V y 3.7 V
<b>Tiempo de carga</b>	1 h – 1.5 h	8 h – 16 h	2 h – 4 h
<b>Ventajas</b>	-Pequeñas -Flexibles -De gran duración	-Bajo coste de fabricación -Suministro de intensidades de corriente muy elevadas	-Pequeñas -Flexibles -De gran duración

Por lo que, para un vehículo aéreo no tripulado, donde el peso y el volumen de la batería, además de la capacidad de almacenamiento y la tasa de descarga, influyen en el rendimiento de vuelo, una batería LiPo o una Li-ion es la mejor opción.

## 2.5 Microcontroladores

La arquitectura estándar de los microcontroladores está basada en la de Von Neumann, en la cual la CPU (Unidad Central de Proceso), está conectada directamente a una única memoria, donde se almacenan las instrucciones del programa y los datos, cuya unidad depende del ancho del bus que comunica la CPU con la memoria. Familias como la Intel 8051 y la Z80 están diseñadas con esta estructura, que era la más común con la aparición de los primeros microcontroladores [18].

En la estructura Von Neumann, la CPU no puede volver a buscar una instrucción en la memoria hasta que no finalice la instrucción actual, al sólo haber un bus de comunicación, haciendo así que la respuesta sea más lenta que usando otras estructuras.

En la Figura 9, se muestra el esquema de la arquitectura de Von Neumann para los microcontroladores, donde se utiliza la misma memoria central para los datos y las instrucciones (casi siempre sólo la RAM):

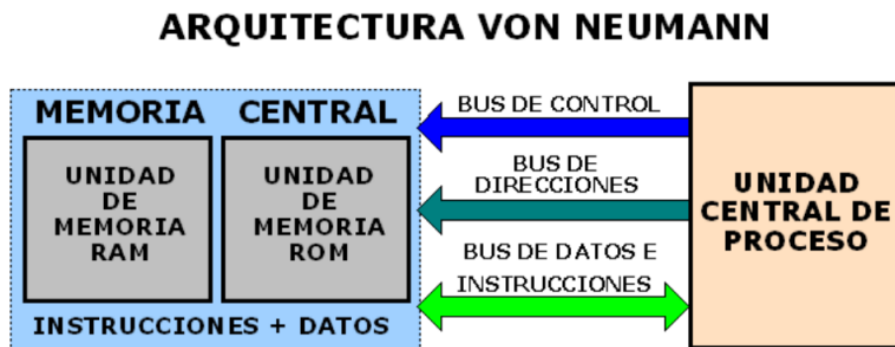


Figura 9. Esquema de la arquitectura Von Neumann.

Otra de las arquitecturas utilizadas es la de Harvard, la más utilizada actualmente en microcontroladores y sistemas integrados, en la cual la CPU está conectada mediante dos buses independientes a dos memorias, una únicamente para el almacenamiento de las instrucciones (Memoria del Programa o ROM) y otra únicamente para el almacenamiento de datos (Memoria de Datos o RAM) [18].

En la Figura 10 se muestra el esquema de la arquitectura Harvard, la cual tiene, por una parte, un bus de instrucciones conectada a una memoria de instrucciones (ROM) y por otra parte, un bus de datos conectada a una memoria de datos (RAM):

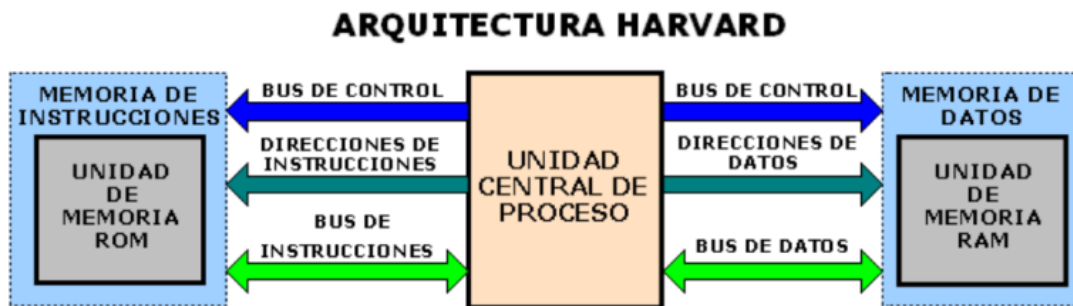


Figura 10. Esquema de la arquitectura Harvard.

Como se observa, esta estructura permite ajustar el tamaño de los buses a las características de cada una de las dos memorias, y también permite al procesador el acceso a las dos memorias de manera simultánea, por lo que la velocidad de procesamiento es más alta que con la estructura Von Neumann [19].

Esta estructura obliga a consumir muchas líneas de entradas y salidas del procesador, pero en los microcontroladores donde los distintos tipos de memoria y el procesador comparten el mismo encapsulado, no supone ningún problema. Es por ello por lo que esta estructura es la más dominante en la gran mayoría de microcontroladores del mercado actual.

Algunos de los chips o dispositivos externos que incluyen los microcontroladores típicos son:

- Generador de reloj, el cual genera una onda cuadrada de frecuencia alta y configura los pulsos de reloj que se usan para sincronizar todas las operaciones del sistema.

- La memoria, que, por una parte, tiene la ROM, la cual es no volátil y almacena el programa de instrucciones, y, por otra parte, la RAM, que es volátil y almacena todas las variables y datos. También existe la memoria EPROM (*Erasable Programmable Read Only Memory* en inglés) que es una memoria programable y borrrable, y la memoria EEPROM (*Electrical Erasable Programmable Read Only Memory* en inglés) y la memoria Flash, que son no volátiles, y que se puede escribir y borrar, además de tener un consumo muy bajo.
- Convertidor analógico a digital (ADC, *Analog to Digital Converter* en inglés), con resoluciones típicas de 8, 10 o 12 bits.
- Convertidor digital a analógico (CAD, *Digital to Analog Converter* en inglés).
- Temporizadores, que permiten controlar periodos de tiempo y llevar la cuenta de contadores.
- Modulador de anchura de pulsos o PWM, que permite proporcionar una salida con anchura variable.
- Perro guardián o Watchdog, un contador que cuando llega a su límite, realiza un reset automático del sistema y que funciona de forma continuada. Esto es útil para detectar fallos en la ejecución, ya que el perro guardián se encarga de reiniciar el sistema en caso de llegar al límite del contador sin que se haya ejecutado ninguna instrucción.
- UARTs (Transmisor-Receptor Asíncrono Universal, *Universal Asynchronous Receiver-Transmitter* en inglés) o USARTs (Transmisor-Receptor Síncrono y Asíncrono Universal, *Universal Synchronous and Asynchronous serial Receiver- Transmitter* en inglés).
- Buses de interfaz serie como I<sup>2</sup>C (desarrollado por Philips y usado en televisores) y CAN (desarrollado por Bosch e Intel y usado en automóviles, permite adaptar con redes de conexionado multiplexado), o buses serie como USB (Universal Serial Bus).

En la Figura 11, se puede observar en conjunto los dispositivos típicos de un microcontrolador mencionados anteriormente:



## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

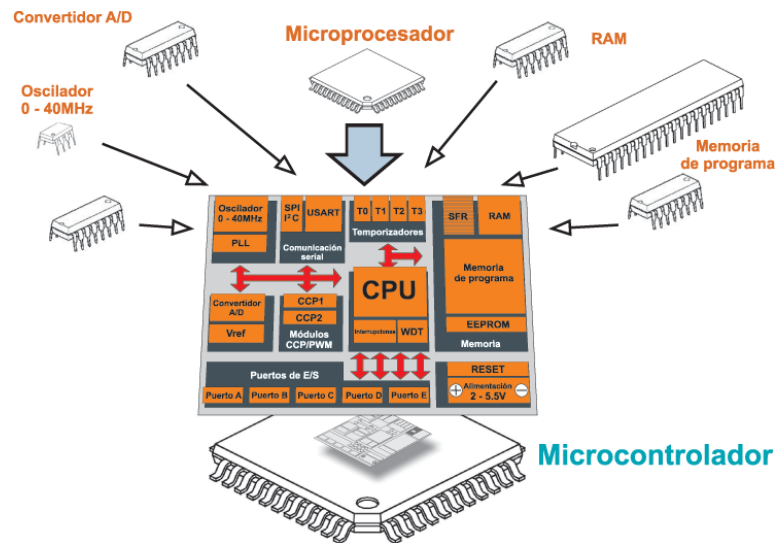


Figura 11. Esquema del conjunto del microcontrolador.

Por lo tanto, el uso de microcontroladores cumple una tarea fundamental en la industria, y es utilizado en la automatización de diversos procesos, como en electrodomésticos, industria informática (como en las impresoras o escáneres), automoción (en el sistema de climatización o el sistema ABS), sistemas de supervisión, ascensores, alarmas...etc.

Algunos de los microcontroladores más populares en el mercado son los siguientes:

- AVR: de la empresa Atmel, con una arquitectura RISC y una velocidad de reloj de hasta 20 MHz, incluye algunos interfaces de comunicación, ADC y PWM.
- Serie STM32F4: de STMicroelectronics, con una arquitectura ARM (Advanced RISC Machine) Cortex-M4F con FPU (*Floating Point Unit* en inglés), una velocidad de reloj de hasta 180 MHz, múltiples periféricos de comunicación, ADC y DAC y controlador gráfico integrado.
- PIC: de la empresa Microchip, con una arquitectura RISC, al igual que AVR, pero con una velocidad de reloj ligeramente más alta, hasta 32 MHz, e incluye varios interfaces de comunicación, ADC y PWM.
- ESP: de la empresa Espressif Systems, con una arquitectura Xtensa LX6, una velocidad de reloj de 240 MHz, múltiples interfaces de comunicación, ADC y DAC, controlador Wi-Fi integrado y múltiples interfaces de conectividad.

Resumiendo, cada uno tiene sus limitaciones y sus ventajas, como el STM32F4, que destaca por su alta velocidad de reloj, su alta variedad de periféricos y su alta seguridad, al igual que su bajo coste, los AVR y PIC, que son buenas opciones para proyectos de menor complejidad, los Arduino, que son bastante populares, o el ESP, que destaca por su controlador Wi-Fi integrado.

La elección del microcontrolador depende de las necesidades del proyecto o aplicación, incluyendo el número entradas/salidas requeridas, el número de periféricos y la conectividad.

### **3. DESCRIPCIÓN DE LA SOLUCIÓN**

En el presente capítulo, se pretende mostrar la solución planteada a los problemas de emisiones de gases de efecto invernadero vinculados al uso de combustibles fósiles y a la poca autonomía que ofrecen los sistemas de generación de energía limpia actuales.

La solución planteada se trata un nuevo sistema de generación de energía para drones que evita la emisión de gases contaminantes o de efecto invernadero, y, además, ofrece una autonomía de vuelo mucho mayor que los drones con baterías únicamente.

Este sistema está formado por una batería de Litio Polímero (batería LiPo) de varias celdas en serie para poder entregar la energía necesaria en los picos de demanda de corriente producidos por el dron en las fases de despegue y aterrizaje, y en paralelo, una pila de combustible de tipo PEM (Membrana de Intercambio Protónico), que consigue alimentar la batería en periodos de vuelo estacionario mediante, por un lado, la introducción Hidrógeno ( $H_2$ ) proveniente de un depósito, y por otro lado, la introducción del Oxígeno ( $O_2$ ) proveniente del aire.

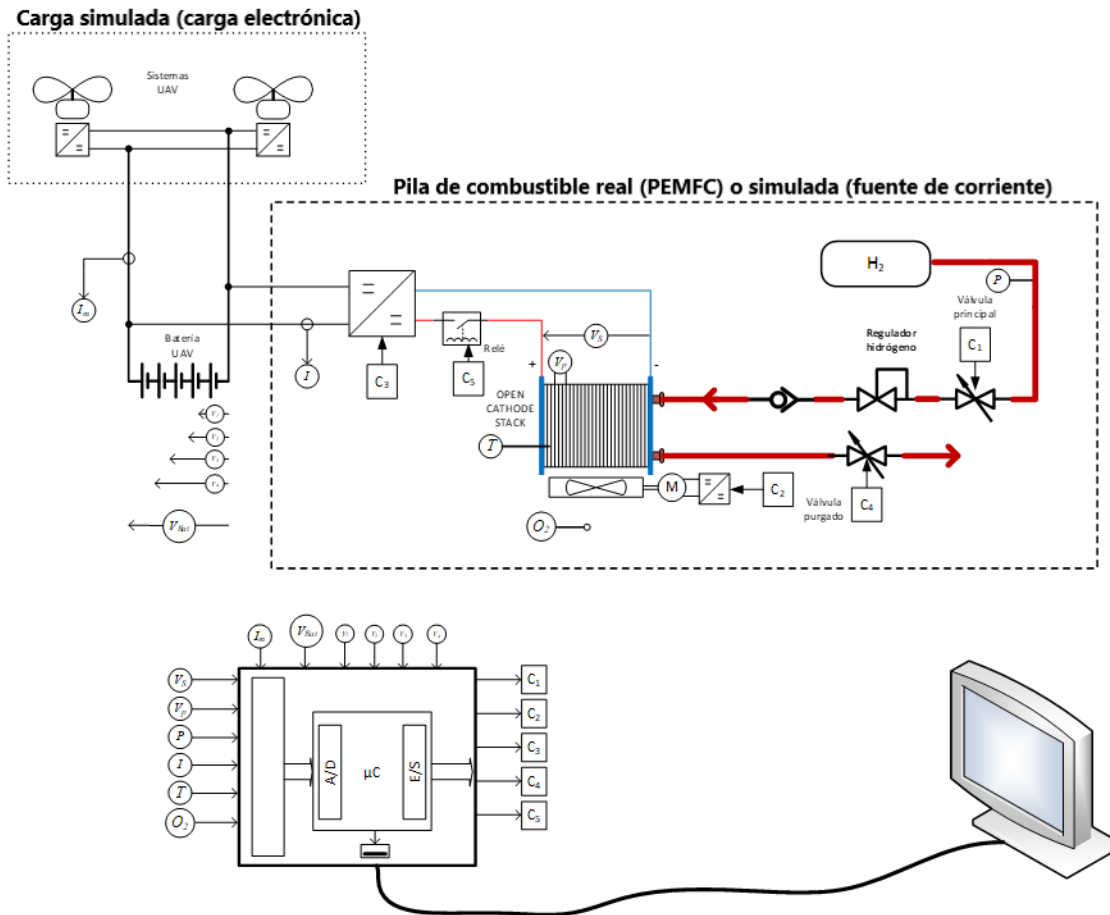
Para actuar sobre la pila de combustible, y controlar y visualizar los consumos de corriente, las tensiones de las celdas de la batería, la temperatura de la pila, el oxígeno que le entra y la presión con la que sale el hidrógeno del depósito, se utilizó un microcontrolador que ejecuta un programa en lenguaje C. En la codificación del software se hizo uso de máquinas de estados finitos para actuar sobre la pila de combustible.

#### **3.2 Arquitectura del sistema**

El modelado del sistema de alimentación formado por una batería y una pila de combustible para drones pretende garantizar una mayor autonomía de vuelo que los sistemas que cuentan únicamente con baterías, y sus componentes se han seleccionado cuidadosamente teniendo en cuenta algunos parámetros importantes para esta aplicación, como los pesos, eficiencia, o efectividad, entre otros.

# Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

En la Figura 12 se observa el modelado propuesto por Marqués Gimeno en su Trabajo de Fin de Máster (TFM) [1], de un sistema de energía para drones formado por una batería y una pila de combustible en paralelo:



**Figura 12.** Esquema representativo del sistema de alimentación "pila de combustible - batería" de un dron o vehículo aéreo no tripulado.

En primer lugar, como se observa, el depósito de hidrógeno ( $H_2$ ) lleva incorporado un manómetro (P) para realizar el control de su nivel, ya que en el depósito la presión disminuye al disminuir el nivel de hidrógeno. Este manómetro será el encargado de mandar una señal de control al dron para realizar un aterrizaje lo más pronto posible cuando el nivel pasa de un mínimo establecido.

A continuación del manómetro, se encuentra la válvula principal que actúa como regulador de presión, y es la encargada de adaptar el suministro de hidrógeno en el compartimento anódico de la pila de combustible, y la válvula de antirretorno, que evita que el flujo de hidrógeno regrese al depósito. Estas dos válvulas se encuentran normalmente incorporados en la mayoría de las pilas de combustible.

La pila de combustible seleccionada es de membrana de intercambio de protones o PEMFC (*Proton-Exchange Membrane Fuel Cell* en inglés), ya que son las que mayor desarrollo presentan en el sector del transporte, por su baja temperatura de operación (30-100 °C), su pequeño tamaño, su elevada eficiencia a la hora de hacer la conversión electroquímica, y la elevada densidad energética del hidrógeno que se utiliza como combustible.

La configuración de la PEMFC seleccionada es en cátodo abierto, que es aquella en la que el cátodo no está haciendo contacto directo con el combustible, y el oxígeno se suministra desde el aire del ambiente. Esta configuración no requiere de sistemas de purificación de oxígeno o compresores de aire, al contrario que las pilas de combustible PEM en cátodo cerrado, por lo que se hace más simple y menos costoso, además, el nivel de contaminación del combustible se reduce, ya que, al no estar en contacto con el cátodo, se evitan reacciones no deseadas [20].

Al seleccionar la configuración del PEMFC en cátodo abierto, es necesario añadir (normalmente viene incorporado) un ventilador, que se acciona con un motor, para captar el aire de la atmósfera, ya que el oxígeno atmosférico es el que actúa como oxidante en la reacción electroquímica al ser suministrado por el cátodo, y como refrigerante de la pila de combustible, cuya temperatura se controla incorporando un sensor de temperatura (T) colocado a la salida del compartimento catódico.

En una situación ideal, en una pila de combustible PEMFC en cátodo abierto, el oxígeno que se suministra por el cátodo se distribuiría de manera homogénea por el compartimento catódico de todas las celdas de la pila de combustible. Pero, en la realidad, se producen pérdidas de carga en los canales de distribución del aire debido a las impurezas que se introducen, y esto genera una distribución no uniforme en las celdas de la pila, por lo que no todas las celdas proporcionarían el mismo valor de tensión.

Por esa razón, existe una válvula de purgado, la cual se acciona cuando se detecta que la tensión de la última celda de la pila o “*Stack*” ( $V_p$ ) es notablemente inferior a un valor tensión de celdas crítico, que se determina a partir de la tensión de salida de la pila ( $V_S$ ) y el número de celdas. Esta válvula permite que una corriente del combustible (hidrógeno) se impulse hacia el exterior, evacuando así las impurezas que se introducen con el aire atmosférico.

A la salida de la pila de combustible o “*Stack*”, se encuentra un relé, que la protege en caso de fallo del convertidor de potencia DC/DC (*Direct Current – Direct Current* en inglés). Este convertidor se encarga de adaptar la tensión de salida de la pila de combustible ( $V_S$ ) a la tensión de la batería de Litio Polímero conectada a la pila, la cual está compuesta por 4 celdas en serie de 3.7 V (Batería LiPo 4S1P, 14.8 V de tensión nominal) está conectada en paralelo a la pila de combustible, para que el dron (carga electrónica) pueda recibir la corriente eléctrica que producen ambos.

Las baterías LiPo incluyen normalmente un conector JST para medir las tensiones de las celdas respecto a la referencia (0 V), y otro conector de enchufe en T de silicona para la carga y descarga de la batería, como se observa en la Figura 13, donde se muestra la batería LiPo que se utilizó, con dichos conectores:



**Figura 13.** Batería LiPo 4S1P empleada (14.8 V de tensión nominal y 3300 mA/h).

La batería LiPo que se emplea tiene una capacidad de almacenamiento de 3300 mA/h (miliamperios hora).

Por otro lado, no se utilizó un dron real para realizar las pruebas, si no que se simuló dicho dron con una carga electrónica, desarrollada en los laboratorios del Departamento de Ingeniería Electrónica (DIE) de la Universidad Politécnica de Valencia (UPV). Esta carga, que se muestra en la Figura 14, es regulable y se comunica con el PC a través de un cable USB y un puerto serie donde puede utilizarse software como Putty, para enviar comandos con el consumo de corriente deseado.

La simulación del dron sirve como fase previa a las pruebas con el dron real, que están fuera del alcance de este TFG.



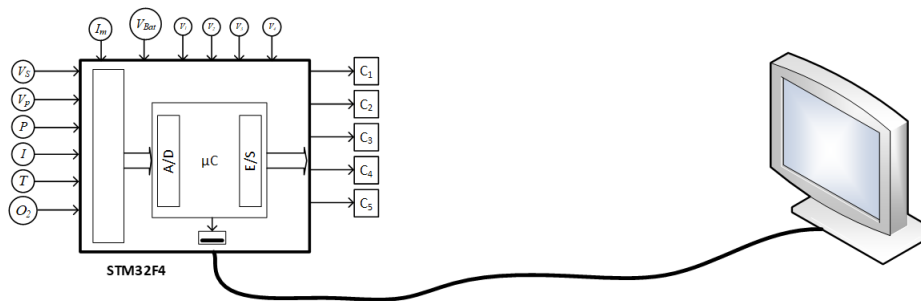
**Figura 14.** Carga electrónica empleada para simular el comportamiento del dron.

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

Como se observa, la carga electrónica, además de la comunicación mediante cable USB, necesita de una tensión de entrada para garantizar su consumo, la cual debe ir a través de dos cables banana.

Para realizar el control o actuar sobre las variables, se empleó un microcontrolador ( $\mu\text{C}$ ), como se observan en la Figura 15, donde se muestra un esquema conceptual de un microcontrolador conectado a un ordenador, con todas las variables conectadas al ADC o a las entradas y salidas digitales:



**Figura 15.** Esquema representativo del microcontrolador conectado al ordenador.

Para ello se seleccionó la placa de desarrollo STM32F429I-DISC1, que se muestra en la Figura 16, con el microcontrolador STM32F429ZIT6, con Procesador Arm® 32-bit Cortex®-M4 CPU, que cuenta con dos canales DAC y tres ADCs con 15 canales, además de 17 temporizadores o timers, y 21 interfaces de comunicación, como UART, USART o I2C.

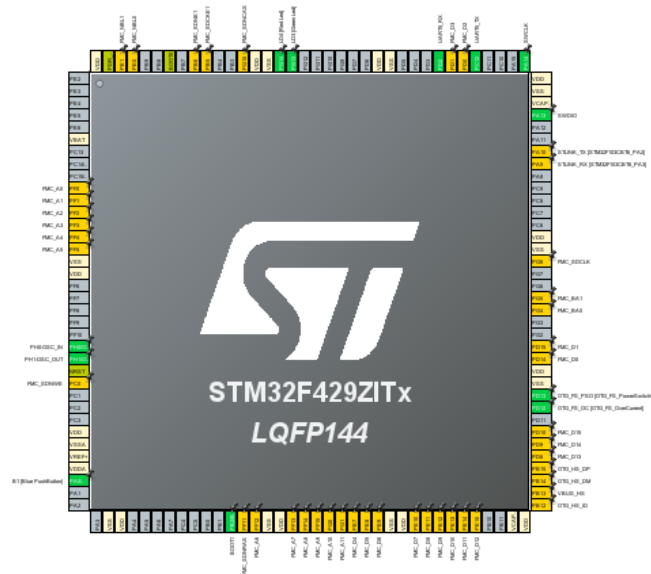


**Figura 16.** Placa de desarrollo STM32F429I-DISC1 con microcontrolador STM32F429ZIT6 [21].



## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

El entorno de desarrollo (IDE) que se empleó para codificar el control fue STM32CubeIDE, que permite la configuración de periféricos en una interfaz como la que se muestra en la Figura 17, para una posterior generación automática del código correspondiente a esas configuraciones.



**Figura 17.** Interfaz de usuario para configurar los pines del microcontrolador STM32F429ZIT6 en la plataforma STM32CubeIDE.

Este microcontrolador se encarga de leer los valores de presión (P), del manómetro, a la salida del depósito de hidrógeno, para controlar su nivel, el flujo de oxígeno ( $O_2$ ) en el aire atmosférico que se introduce, la temperatura de la pila (T), la tensión de salida que proporciona la pila o “Stacks” ( $V_S$ ), la tensión de celda ( $V_P$ ), la corriente de salida de la pila (I), el consumo de corriente del dron o carga electrónica ( $I_m$ ), las 4 tensiones de celda de la batería LiPo ( $V_1, V_2, V_3, V_4$ ), las cuales normalmente no están perfectamente balanceadas, y la tensión total de la batería ( $V_{BAT}$ ), que es la misma que la tensión de la cuarta celda. Estas lecturas se realizan mediante el uso de varios canales ADC (*Analog to Digital Converter* en inglés).

Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

También se va a encargar de actuar sobre algunas variables, como la apertura o cierre de la válvula principal ( $C_1$ ), el accionamiento del motor del ventilador para permitir la entrada de aire atmosférico ( $C_2$ ), el disparo del convertidor DC/DC ( $C_3$ ), la apertura o cierre de la válvula de purgado ( $C_4$ ), y el accionamiento del relé de seguridad ( $C_5$ ). Sobre estas variables se actuará utilizando E/S (Entradas y Salidas) digitales y canales DAC (*Digital to Analog Converter* en inglés).

Por lo tanto, se dispone de 12 variables que requieren de lectura (adquisición) y visualización en pantalla, y 5 variables de escritura, como se resume en la siguiente Tabla 5:

**Tabla 5.** Clasificación y descripción de las variables del sistema "batería-pila de combustible" del dron.

Variable	Descripción de la variable	Función del microcontrolador
$V_1$	Tensión de 1º celda de la batería LiPo	Input o entrada
$V_2$	Tensión de 2º celda de la batería LiPo	Input o entrada
$V_3$	Tensión de 3º celda de la batería LiPo	Input o entrada
$V_4$	Tensión de 4º celda de la batería LiPo	Input o entrada
$V_{BAT}$	Tensión total de la batería LiPo	Input o entrada
$I_m$	Corriente de consumo del dron	Input o entrada
I	Corriente de salida de la PEMFC	Input o entrada
$V_p$	Tensión de celda	Input o entrada
$V_s$	Tensión de salida de la pila o “Stacks”	Input o entrada
T	Temperatura de la PEMFC	Input o entrada
$O_2$	Flujo de oxígeno	Input o entrada
P	Presión del depósito de hidrógeno	Input o entrada
$C_1$	Válvula principal	Output o salida
$C_2$	Convertidor DC/DC del ventilador	Output o salida
$C_3$	Convertidor DC/DC de la pila	Output o salida
$C_4$	Válvula de purgado	Output o salida
$C_5$	Disparo del relé de seguridad	Output o salida

### 3.3 Diseño del hardware

Previo a la programación, fue necesario comprender qué señales va a leer el microcontrolador y como se va a comunicar con el PC para visualizar esos datos, además de implementar los circuitos de adaptación de las señales que se desean medir.

Una cuestión importante para tener en cuenta es que, según la hoja de especificaciones del microcontrolador, los canales del convertidor analógico a digital (ADC) trabajan en un rango de 0 V a 3.3 V, y una tensión mayor a 3.6 V podría ser dañino para el microcontrolador.

Por lo que, por una parte, había que realizar el estudio de los rangos de voltaje de las señales que se desean medir, como los consumos de corriente o las tensiones de la batería, adaptarlas al rango con el que el ADC trabaja y seleccionar los canales ADC a utilizar.

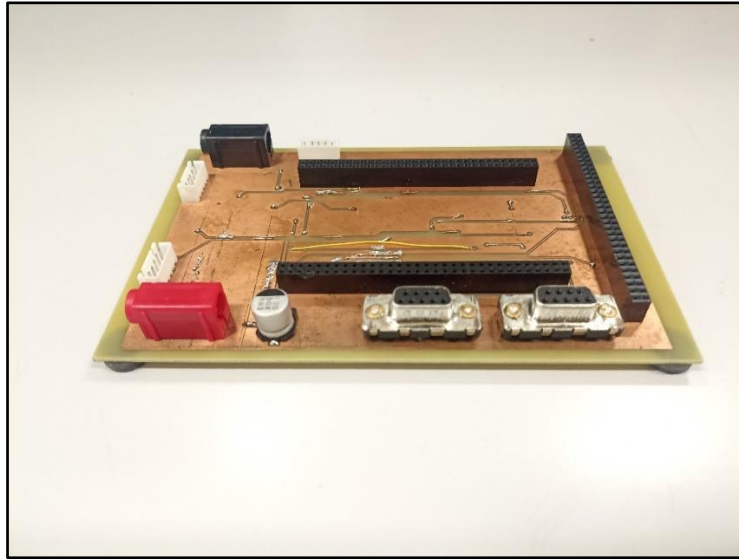
Por la otra parte, fue necesario diseñar la comunicación con el PC, para la cual se va a usó el estándar RS232, que es una comunicación sencilla por cable, perfecta para distancias cortas, y que permite una velocidad de transmisión de hasta  $1.152 \cdot 10^5 \frac{\text{bits}}{\text{segundo}}$  [22], a través de la UART. Es necesario un dispositivo que convierta los niveles de voltaje de TTL/CMOS (niveles lógicos digitales usados comúnmente en microcontroladores y usado por la UART) a los niveles de voltaje que se utilizan en el estándar RS232.

Por todo ello, fue necesario realizar varias adaptaciones de señal y, la solución planteada requirió el diseño de una placa de circuito impreso con dos conectores GPIO montados en paralelo, como el que se muestra en la Figura 22, para introducir el microcontrolador y conseguir una interfaz con el mismo y con otros tipos de conectores para los diferentes sensores, la batería, o la comunicación UART, con sus correspondientes adaptaciones.

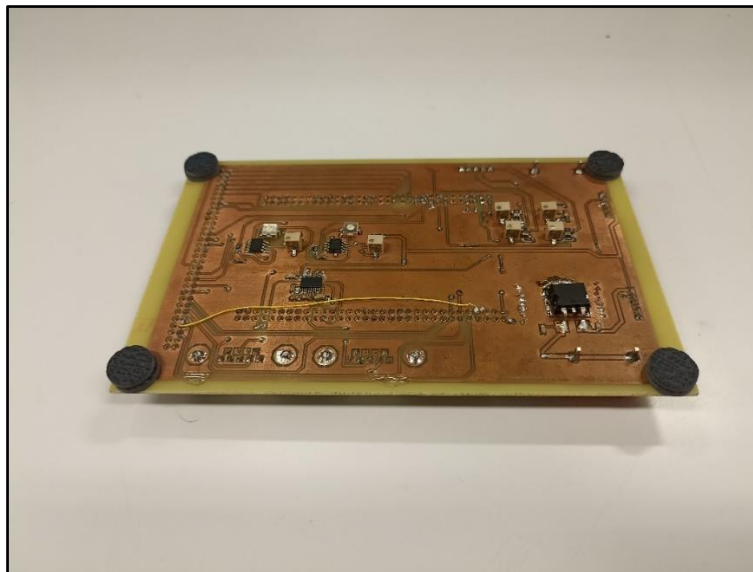
La tarjeta de interfaz de hardware diseñada se muestra en las Figuras 18 y 19:

Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---



**Figura 18.** Parte superior (top) de la tarjeta de interfaz de hardware diseñada.



**Figura 19.** Parte inferior (bottom) de la tarjeta de interfaz de hardware diseñada.

Como se observa, la parte superior está compuesta por todos los conectores necesarios, mientras que en la parte inferior se encuentran los elementos necesarios para los acondicionamientos de las señales.

Al no disponer de una pila de combustible real durante la realización de mi TFG, este dispositivo se simuló con una fuente de alimentación del laboratorio del DIE en la UPV, y en la placa hardware de interfaz se incluyó un tercer conector GPIO, que se muestra en perpendicular a los otros dos en la Figura 18 al cual se le han conectado las entradas y salidas digitales, los canales ADC, y los canales DAC del microcontrolador que estarían conectados a la pila de combustible.

La simulación consistió en generar las tensiones de los sensores con una fuente de alimentación, y con varios potenciómetros para regular dichas tensiones en una ProtoBoard, que simularon las señales eléctricas a leer por el microcontrolador de los diferentes sensores o tensiones de salida que tendría la pila real, a través de los conectores GPIO preparados para la pila de combustible. También se simularon las variables sobre las que actuar con salidas digitales y canales DAC, y se midieron y visualizaron a través de multímetros.

La pila de combustible real que se empleará en un futuro será el modelo H100 de la marca h2planet, y se muestra en la Figura 20:



**Figura 20.** Pila de combustible, modelo H100 de la marca h2planet [23].

Esta pila de combustible es de tipo PEM y está formado por un “*stack*” de 20 celdas que proporcionan una potencia de 100 W. Su tensión y corriente nominal de salida es de 12 V y 8.3 A, y lleva incorporado el sistema de ventilación, el depósito de hidrógeno, y todos los sensores, convertidores y válvulas mencionadas anteriormente en la arquitectura del sistema. La máxima temperatura del “*stack*” es de 65 °C y su eficiencia eléctrica es del 40% a máxima potencia.

Cabe destacar también que el diseño del esquema eléctrico, las simulaciones del sistema, y el diseño de la PCB de interfaz se realizaron en el programa de simulación Proteus, cuya licencia la posee el DIE en la UPV en sus correspondientes laboratorios.

### 3.3.1 Batería

En primer lugar, fue necesario saber cuántas señales hay que leer, y cuál es su rango de voltaje, para así poder realizar una atenuación o una amplificación de dicha señal.

La batería LiPo que se utilizó, tiene una tensión nominal de 14.8 V, y está formada por 4 celdas en serie, cuya tensión nominal es de 3.7 V, y cuya tensión máxima de celda es de 4.2 V cuando están equilibradas. Hay que tener en cuenta, que, una carga o una descarga desequilibrada, una temperatura desigual, o un daño mecánico pueden provocar situaciones en las que las tensiones de celda se desequilibran, llegando a tensiones de hasta 2 V por encima de su carga máxima. Por lo que, el primer paso fue cargar la batería a su máxima capacidad, y medir las tensiones de celda en esa situación, a través del conector JST de 5 pines, y los resultados son los siguientes:

- 1º celda → 4.13 V
- 2º celda → 8.29 V
- 3º celda → 12.44 V
- 4º celda → 16.59 V

Como se observa, las tensiones de celda son acumulativas, es decir, la segunda celda proporciona la tensión de la primera y la segunda, y así sucesivamente hasta la cuarta celda.

El siguiente paso fue el diseño de un circuito atenuador de tensión, pero teniendo en cuenta que cada tensión de celda puede llegar a aumentar hasta 2 V, para conseguir un rango con el que el ADC pueda trabajar, el cual se ha decidido que será entre 0 V y 2.7 V, para detectar sobretensiones.

El circuito que se diseñó para ello está formado por divisores de tensión, que se encargan de atenuar cada máxima tensión de celda a un valor de 2.7 V

aproximadamente, y la fórmula del divisor de tensión que se empleó para calcular las resistencias a colocar es la siguiente:

$$V_{in(máx)} = V_{out(máx)} \cdot \frac{R_1 + R_2}{R_2} \text{ (V)} \quad (\text{Ec.1})$$

Despejando  $R_1$ :

$$R_1 = \frac{V_{in(máx)} \cdot R_2}{V_{out(máx)}} - R_2 \text{ (}\Omega\text{)} \quad (\text{Ec.2})$$

Siendo  $V_{in(máx)}$  la máxima tensión de entrada que se quiere atenuar,  $V_{out(máx)}$  la máxima tensión de salida que se desea obtener, que es 2.7 V,  $R_1$  la resistencia conectada al terminal positivo de la tensión de entrada, y  $R_2$  la resistencia conectada al terminal negativo.

Al ser  $R_2$  más relevante, se decidió que sean potenciómetros, para poder así ajustar con precisión las resistencias y compensar las tolerancias de estas, consiguiendo así el valor de tensión deseado. También se decidió que las resistencias a emplear sean normalizadas en la serie E12 de la norma IEC, ya que son fáciles de encontrar en el mercado, son económicas y tienen una buena precisión, con una tolerancia del  $\pm 10\%$ , la cual es suficiente para la mayoría de las aplicaciones.

Por lo tanto, aplicando la Ecuación 2 para cada una de las celdas, y suponiendo un valor de  $R_2$ , que será un potenciómetro ajustable de 1 k $\Omega$ , se obtuvieron las siguientes resistencias normalizadas en la serie E12:

Para la 1° celda, suponiendo un valor de 780  $\Omega$  para el potenciómetro:

$$R_1 = \frac{6.13 \cdot 780}{2.7} - 780 = 990.88 \Omega \approx 1000 \Omega$$

Para la 2° celda, y suponiendo un valor de 960  $\Omega$  para el potenciómetro:

$$R_3 = \frac{10.29 \cdot 960}{2.7} - 960 = 2698.66 \Omega \approx 2700 \Omega$$

Para la 3° celda, y suponiendo un valor de 760  $\Omega$  para el potenciómetro:

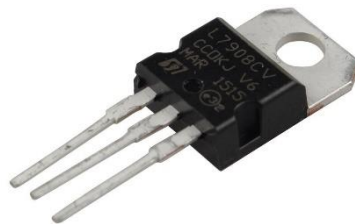
$$R_5 = \frac{14.44 \cdot 760}{2.7} - 760 = 3304.6 \Omega \approx 3300 \Omega$$

Para la 4° celda, y suponiendo un valor de 960  $\Omega$  para el potenciómetro:

$$R_7 = \frac{18.59 \cdot 560}{2.7} - 560 = 3295.7 \Omega \approx 3300 \Omega$$

Siendo  $R_1$  la resistencia conectada al terminal positivo de la 1° celda,  $R_3$  la de la 2° celda,  $R_5$  la de la 3° celda, y  $R_7$  la de la 4° celda, todas normalizadas en la serie E12 según la norma IEC.

La tensión de la 4° celda corresponde a la tensión total de la batería ( $V_{BAT}$ ), y se decidió utilizar esta tensión para alimentar el resto de los componentes de la placa de adaptaciones, como, por ejemplo, el propio microcontrolador, conectando un regulador lineal bastante común, el L7805C, como el que se observa en la Figura 21, con el conector de enchufe en T de la batería LiPo. Este regulador lineal proporciona una tensión de 5 V estable (tensión de alimentación bastante común en muchos componentes) y una corriente de 1 A, a partir de una tensión de entrada de entre 8 V y 30 V.



**Figura 21.** Regulador lineal L7805C.

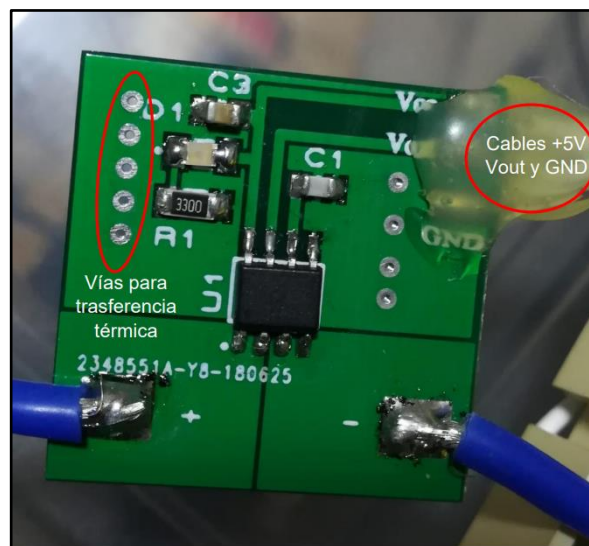
Se utilizó un conector banana de prueba para conectar la batería a la placa, y un condensador electrolítico 470  $\mu\text{F}$  en paralelo, además de los dos condensadores cerámicos que el fabricante recomienda conectar en paralelo para mejorar la señal y hacerla más estable, de 100 nF y 330 nF.



La función del condensador electrolítico es evitar las caídas de tensión que se producen cuando hay grandes picos de consumo de corriente, ya que el cableado actúa como bobina de una sola espira en ese caso y se produce una caída importante de la tensión, pudiendo provocar un reinicio del sistema.

### 3.3.2 Sensor de corriente

Como se ha mencionado anteriormente, el dron se simuló con una carga electrónica regulable, y la pila de combustible con una fuente de alimentación del laboratorio, y, se decidió que la máxima corriente de consumo de la carga era de 4 A, al igual que la corriente de salida de la fuente de alimentación que simuló la pila. Para leer dichas corrientes, se utilizaron los sensores de corriente ACS712, que se fueron integrados en una placa de circuito impreso, con los componentes que el fabricante recomienda conectar. El diseño de esta placa de circuito impreso con el sensor de corriente ACS712 ha sido realizado por Muñoz en su TFG [24] y se muestra en la Figura 22.



**Figura 22.** Sensor de corriente ACS712 integrado en placa de circuito impreso [24].

Este circuito integrado basado en un sensor de efecto Hall, tiene las siguientes características:

- Tamaño muy reducido.
- Medida de hasta 30 A.
- Medida de corriente continua (DC).

- Señal de salida analógica.
- Tensión de alimentación de 5 V.

El ACS712, al detectar 0 A, la tensión de salida es de 2.5 V (offset inicial), ya que es capaz de medir corrientes negativas, y, según su hoja de especificaciones, su sensibilidad es de 66 mV/A, por lo que, al medir los 4 A de máxima corriente, el sensor proporciona la siguiente tensión de salida:

$$V_{Sensor} = 2.5 + I \cdot Sensibilidad (V) \quad (Ec.3)$$

$$V_{Sensor} = 2.5 + 4 \cdot 0.066 = 2.764 V$$

Siendo I la corriente a medir que pasa por el sensor, se obtiene una tensión de salida de 2.764 V.

Para no perder resolución, se decidió restar el offset a la salida del sensor para obtener únicamente la medida real en función de la corriente a medir, por lo que, la tensión de salida al detectar 4 A de corriente quedaría:

$$V_{Sensor \text{ sin offset}} = 2.764 V - 2.5 V = 0.264 V$$

Esta tensión necesita una amplificación para adaptarla a los rangos de tensión de trabajo del ADC del microcontrolador, que se ha decidido que trabajará de 0 V a 2.7 V.

Para la amplificación y la resta se diseñó un circuito con un amplificador de instrumentación (AI), el modelo AD623 de la empresa Analog Devices, un amplificador con salida “rail-to-rail”, que permite trabajar con voltajes muy cercanos a los de alimentación [25] y que permite una tensión de alimentación bipolar o unipolar.

Considerando esto, se decidió trabajar con una alimentación unipolar, con:

- Tensión de alimentación negativa ( $-V_s$ ) = 0 V.
- Tensión de alimentación positiva ( $+V_s$ ) = 5 V provenientes del regulador lineal L7805C.

A este amplificador de instrumentación, se le introduce por el terminal no inversor la tensión de salida del sensor con offset ( $V_{Sensor}$ ), y por el terminal inversor se le introduce un divisor de tensión, que divide en dos una tensión de 5 V proveniente del regulador lineal L7805C.

Este divisor está formado por una resistencia de  $470\Omega$  y un potenciómetro de  $1\text{ k}\Omega$ , para ajustar con precisión las resistencias y compensar la tolerancia que presentan estas, para que queden con el mismo valor y den como resultado una división precisa de la tensión.

El siguiente paso fue calcular la resistencia de ganancia (RG), la cual proporciona la ganancia deseada, que, en este caso, fue:

$$Ganancia = \frac{V_{out,m\acute{a}x} (V)}{V_{in,m\acute{a}x} (V)} = \frac{2.7\text{ V}}{0.264\text{ V}} = 10.23 \frac{V}{V} \quad (Ec.4)$$

Y la resistencia de ganancia (RG), según la hoja de especificaciones del AD623, se calcula de la siguiente manera:

$$RG = \frac{100\text{ k}\Omega}{Ganancia-1} = \frac{100\text{ k}\Omega}{10.23-1} = 10.83\text{ k}\Omega \quad (Ec.5)$$

Sabiendo el valor de la resistencia de ganancia, se decidió que será también un potenciómetro, concretamente de  $20\text{ k}\Omega$ , para poder ajustar con precisión la tensión de salida del AI, y compensar las tolerancias que presentan las resistencias.

### 3.3.3 UART

Para la recepción y visualización de datos en el ordenador, como, por ejemplo, las gráficas de consumo de corriente del dron o las tensiones de celda de la batería, es necesario diseñar la comunicación serie adecuada.

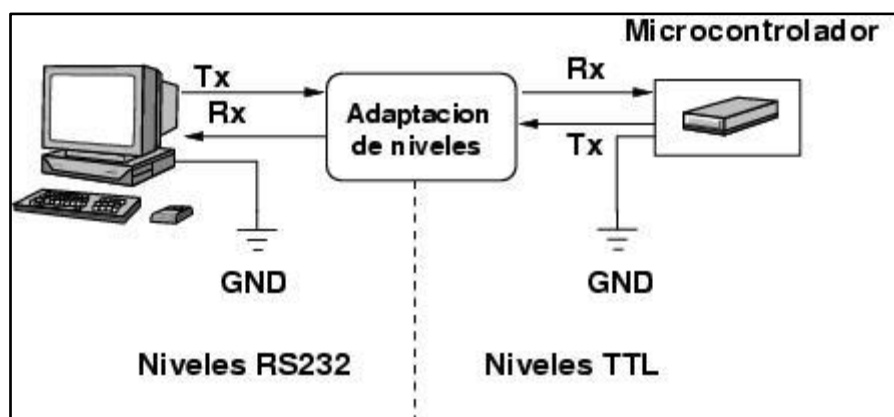
Como se ha mencionado anteriormente, el estándar de comunicación seleccionado fue RS232, un estándar de comunicación serial por cable establecido en 1960, en el que las señales de transmisión (Tx) y recepción (Rx) suelen estar entre  $\pm 5\text{ V}$  y  $\pm 25\text{ V}$  [26]. Este estándar utiliza conectores DB-9, es menos susceptible a ruidos y puede funcionar con longitudes de cable de hasta 15 metros.

El dispositivo electrónico del microcontrolador encargado de transmitir y recibir datos fue UART (*Universal Asynchronous Receiver-Transmitter* en inglés). La UART convierte los datos paralelos que se transmiten dentro del microcontrolador en datos seriales que pueden ser enviados a través de un cable.

La comunicación UART es asíncrona, es decir, no hay un reloj externo que sincronice constantemente la transmisión y recepción de los datos [27], si no que utiliza el bit de inicio y el bit de parada para indicar el comienzo y el fin de cada transmisión. Por esto, la comunicación UART no es tan precisa como la I2C, por ejemplo, pero es una forma sencilla y bastante económica de transmitir y recibir datos seriales (en distancias cortas).

Al ser asíncrona, el receptor y el transmisor deben realizar la transmisión a la misma velocidad, la cual se mide en baudios, e indican cuántas veces se transmite la señal en un solo segundo. Los valores más típicos son 9600, 19200, 38400, 57600 o 115200 baudios.

Tras la conversión de datos en la UART, fue necesario un dispositivo que convierta de niveles de voltaje de los datos seriales de la UART a niveles de voltaje compatibles con el estándar RS232, para transmitir y recibir datos desde el ordenador al microcontrolador, como se observa en la Figura 23.



**Figura 23.** Esquema conceptual de la transmisión y recepción de datos desde el microcontrolador al PC y viceversa.

El dispositivo más adecuado para ello es el MAX13223EEUP+, más conocido como MAX232, un circuito integrado bastante económico que dispone de cuatro puertos de entrada o salida (dos de transmisión y 2 de recepción), y convierte de niveles TTL o CMOS (0 V a 5 V) en señales de voltaje RS232, y viceversa.

Este dispositivo puede ser alimentado con una tensión de 3 V a 5.5 V, y, según la hoja de especificaciones del dispositivo, los valores de los condensadores que el fabricante recomienda conectar dependen de esa tensión de alimentación, como se observa en la Tabla 6:

**Tabla 6.** Valores de los condensadores a conectar al MAX232 según la tensión de alimentación.

$V_{CC}(V)$	C1 ( $\mu F$ )	C2, C3 ( $\mu F$ )	C4 ( $\mu F$ )
De 3.0 a 3.6	0.1	0.1	1
De 4.5 a 5.5	0.047	0.33	1

Siendo  $V_{CC}$  la tensión de alimentación en Voltios (V), se seleccionaron los valores de los condensadores con la tensión de alimentación de 4.5 V a 5.5 V, ya que este dispositivo es alimentado con 5 V por el regulador L7805C que proviene de la batería LiPo. También se conectó un condensador de 100 nF en la entrada de alimentación del MAX232, que actúa como filtro paso bajo, y para evitar que el ruido afecte al funcionamiento del dispositivo.

Cabe destacar que las conexiones a masa del MAX232, corresponden a la masa digital, que está separada de la masa analógica (masa conectada al resto de componentes con señales analógicas). Esto es debido a que este dispositivo trata con señales de alta velocidad, y realizando la separación, se evitan problemas de interferencias o ruido eléctrico, que pueden causar errores en la medición o en la transmisión de datos.

Las masas analógicas y las digitales, a pesar de separarlas, deben estar unidas en un punto, para proporcionar una referencia común para las señales analógicas y digitales [28].

Normalmente, las conexiones en el estándar RS232 se realizan con conectores DB-9, ya que este conector tiene una configuración fija y robusta, por lo que la conexión se hace más confiable, y por su fácil identificación de los pines a conectar.

En primer lugar, la conexión entre el MAX232 y el PC se realizó mediante un conector DB-9 hembra del cual se colocaron 2, para tener uno auxiliar en caso de que uno falle.

A este conector DB-9 hembra se le conectó un cable con conectores DB-9 hembra a DB-9 macho. La conexión macho-hembra se realizó conectando el pin de recepción (Rx) con el de transmisión (Tx).

Finalmente, para la conexión desde el cable con conector DB-9 macho al PC, se utilizó un adaptador DB-9 hembra a USB, como el de la Figura 24:



**Figura 24.** Adaptador de DB-9 a USB para la conexión con el PC.

### **3.3.4 Filtro Paso Bajo (FPB)**

Por último, a la entrada de los canales del ADC, se implementó un filtro para evitar la aparición de “aliasing” en las señales de entrada que superan la frecuencia de Nyquist.

El fenómeno del “aliasing” ocurre cuando en el ADC se muestrean señales con componentes de alta frecuencia, que pueden reflejarse y superponerse en las frecuencias más bajas, creando interferencias en la señal original.

Nyquist establece en su teorema de muestreo, que la frecuencia de muestro debe ser al menos dos veces mayor que la frecuencia a muestrear más alta. Por lo que, un filtro paso bajo (FPB) o filtro “anti-aliasing”, que atenúa las componentes de frecuencias mayores a la frecuencia de corte establecida, evita la distorsión de la señal y reduce el ruido de alta frecuencia, que puede afectar la precisión de las conversiones del ADC [29].

La fórmula del filtro paso bajo (FPB) es la siguiente[30]:

$$f_c = \frac{1}{2\pi \cdot R \cdot C} \text{ (Hz)} \quad (\text{Ec.6})$$

Siendo  $f_c$  la frecuencia de corte, la cual debe ser al menos dos veces mayor que la frecuencia más alta a muestrear.

Se seleccionó una frecuencia de corte de 1 kHz, ya que es una frecuencia media que evita que se atenúen los cambios bruscos de menos de 1 ms y que consigue atenuar las altas frecuencias de ruidos o interferencias. El valor del condensador que se ha seleccionado es de 330 nF, que es un valor bastante común, por lo que, despejando R:

$$R = \frac{1}{2\pi \cdot f_c \cdot C} = \frac{1}{2\pi \cdot 1 \cdot 10^3 \cdot 330 \cdot 10^{-9}} = 482.29 \Omega \approx 470 \Omega$$

Como se observa, el valor exacto es de  $482.29 \Omega$ , el cual se aproxima al valor normalizado en la serie E12 de  $470 \Omega$ .

### 3.4 Diseño del software

Tras el diseño de la placa de interfaz de hardware, el siguiente paso fue codificar el control del dron híbrido. Para ello, había que seleccionar una lógica de control que se adapte a las necesidades del proyecto.

### 3.4.1 Lógica de Control

En la programación existen diferentes estructuras de control, las cuales determinan el orden de ejecución de las instrucciones de un programa o software. Este orden de ejecución se puede expresar mediante un diagrama de flujo o flujograma, que describe la operación del sistema diseñado.

Para diseñar un sistema digital, tras formular el algoritmo (secuencia de pasos) adecuado mediante un diagrama de flujos, hay que seleccionar una lógica de control o método de programación.

En la actualidad, la gran mayoría de ingenieros o informáticos trabajan con alguna de las siguientes lógicas de control:

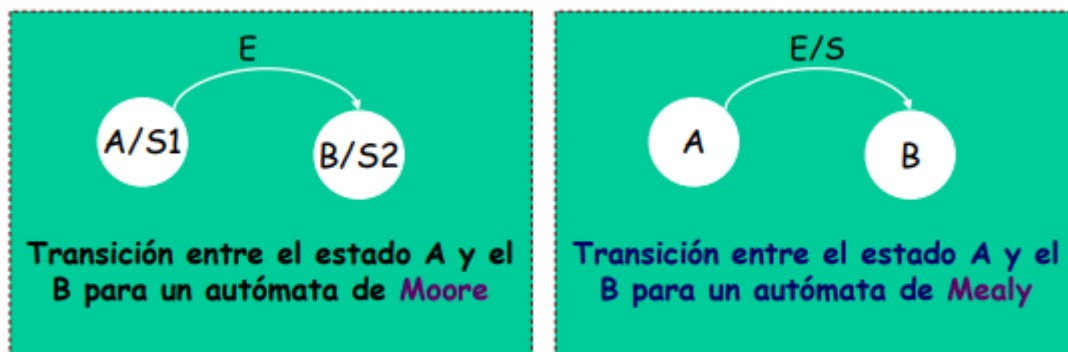
- Plano Secundario y Plano Principal, siendo el plano secundario el bucle sin fin del main() (normalmente es la función que sirve de punto de partida) y el plano principal las interrupciones del sistema. Es una programación manejada por eventos.
- SuperLazo o SuperLoop, es un método complejo y difícil de depurar, donde todo el código está dentro del bucle sin fin del main().
- Máquina de Estados Finita (MEF o *Finite-State Machine* en inglés), que es un modelo matemático formal para modelar sistemas secuenciales, y destaca por reducir la complejidad del código, dividiendo la lógica en bloques discretos, que facilitan la depuración.

Este último método de programación y sus correspondientes diagramas representan el comportamiento dinámico de un sistema reactivo (respuesta a un evento) y se describen como un modelo matemático [31].



Las MEF suelen estar definidos por unos estados (situaciones en las que el sistema puede encontrarse, que pueden ser la espera a otro evento o la ejecución de una actividad), entradas (también llamadas eventos y producen los cambios de estado), salidas (también llamadas acciones y son las respuestas ante los eventos) y las funciones de transición de estado (describen para cada estado, cuál es el siguiente y que acción se realiza cuando se producen los distintos eventos).

La manera de representar una Máquina de Estados Finita o un autómata es mediante un Diagrama de Estados, la cual, como se observa en la siguiente Figura 25, es posible realizar con el método de Moore, donde la salida está definida únicamente por el estado, o con el método de Mealy, donde la salida está definida por el estado y las entradas:



**Figura 25.** Diagramas de Estado de Moore y de Mealy.

Por lo tanto, para implementar una Máquina de Estados Finita en lenguaje C, los pasos a seguir son:

1. Realizar un diagrama de flujo para visualizar los pasos o instrucciones del programa en general.
2. Identificar los eventos (entradas), las acciones (salidas), los estados (se identifican a partir de los requerimientos del problema).
3. Realizar un diagrama de estados, para representar los estados, entradas y salidas del sistema, con las transiciones y las funciones de transición de estado.

4. Codificar en lenguaje C el sistema diseñado, mediante varios métodos, entre ellos el uso de la sentencia “if – else” (si hay un gran número de estados se complica mucho el código), la sentencia “Switch” (la cual hace más fácil de entender el código), o un bucle “for” si hay un número de estados fijos.

Al ser un sistema reactivo (las salidas se activan según los eventos ocurridos), se decidió emplear máquinas de estados finitos para el control del sistema de energía del dron híbrido. La metodología utilizada para la implementación de la máquina de estados finitos de este trabajo ha sido extraída del libro titulado '*Introduction to Embedded Systems*' [32].

### 3.4.2 El control

La codificación realizada consistió en la implementación de una Máquina de Estados Finita (MEF) para el control de las variables de entrada o salida de la pila de combustible mostradas en la Tabla 5.

Esta máquina de estados implementa las siguientes condiciones:

- Si la diferencia entre la corriente de salida de la PEMFC (corriente de carga de la batería,  $I$ ) y la corriente de consumo del dron ( $I_m$ ) es mayor que cero ( $(I - I_m) > 0 A$ ) la batería está en carga.
- Una vez en carga, si la tensión total de la batería LiPo es mayor que la tensión máxima de carga ( $V_{BAT} > 16.59 V$ ), entonces el convertidor DC/DC de la salida de la pila ( $C_3$ ) debe de interrumpir la corriente de carga e igualar la corriente de salida a la corriente de consumo ( $(I - I_m) = 0$ ).
- Si la batería está en carga y la tensión total ( $V_{BAT}$ ) es menor a la tensión máxima de carga ( $V_{BAT} < 16.59 V$ ), se debe asegurar que las tensiones de las cuatro celdas de la batería ( $V_1, V_2, V_3, V_4$ ) estén equilibradas para poder continuar, y si no lo están, se debe interrumpir la corriente de carga igualando la corriente de salida del convertidor DC/DC a la corriente de consumo ( $(I - I_m) = 0$ ).

Para la comprobación del equilibrio entre celdas de la batería LiPo se debe cumplir la siguiente expresión:

$$V_X = \frac{V_{BAT}}{4} \pm 10\% \text{ (V)} \quad (\text{Ec.7})$$

Siendo X el número de celda de la batería LiPo,  $V_{BAT}$  la tensión total de la batería y el 10% el porcentaje de desequilibrio permitido. Las tensiones de celda se han acondicionado para que no superen un valor de 2.7 V en la lectura del microcontrolador.

- La válvula principal del PEMFC ( $C_1$ ) trabaja con una señal todo/nada (1 corresponde a la válvula abierta y 0 corresponde a válvula cerrada) y debe permanecer abierta, excepto si ocurre alguno de los siguientes eventos:
  1. La presión que marca el manómetro de la salida del depósito de hidrógeno es mayor o igual que la presión máxima de seguridad establecida en 0.6 bar ( $P > P_{m\acute{a}x}$ ). En el presente TFG dicho sensor se ha simulado con una fuente de alimentación con un potenciómetro, que genera la señal eléctrica que entregaría el manómetro con una PEMFC real tras un acondicionamiento de la señal, en el que 1 bar corresponde a 2.7 V y 0 bar corresponde a 0 V.
  2. La corriente de salida del PEMFC (cuyo valor nominal es de 8.3 A) supera o es igual a la corriente de salida máxima de seguridad establecida en 10 A ( $I > I_{m\acute{a}x}$ ). Hay que tener en cuenta que los sensores de corriente empleados entregan una tensión alrededor de 0.7 V por Amperio tras su acondicionamiento de señal en la placa de interfaz de hardware diseñada, y que, al simular la PEMFC mediante una fuente de alimentación y potenciómetros se ha decidido que la corriente máxima de seguridad va a ser de 3 A y la corriente nominal de 2 A. Como se ha mencionado anteriormente en el apartado del diseño del acondicionamiento de señal de los sensores de corriente, una lectura de corriente de 4 A corresponde a una tensión de 2.764 V en la salida.

3. La tensión de salida de la pila es superior a la tensión de salida máxima de seguridad con un valor de 24 V o inferior a la tensión mínima de seguridad con un valor de 10 V ( $V_S > V_{m\acute{a}x}$  o  $V_S < V_{min}$ ). En el presente TFG, la tensión de salida de la pila se ha entregado mediante una fuente de alimentación y un potenciómetro, que simula su señal eléctrica en la que 25 V de tensión de salida de la pila a corresponden a 2.7 V en la entrada del microcontrolador y 5 V de tensión de salida de la pila a corresponden a 0.54 V en la entrada del microcontrolador tras un acondicionamiento de la señal.
  4. La temperatura de la PEMFC es superior a 65 °C, que corresponde a la máxima temperatura de la pila real ( $T^{\circ} > T^{\circ}_{m\acute{a}x}$ ). La señal eléctrica del sensor de temperatura de la pila real se ha entregado mediante una fuente de alimentación y un potenciómetro, en la que 70 °C de temperatura en la pila real corresponden a 2.7 V en la entrada del microcontrolador, y 30 °C de temperatura en la pila real corresponden a 1.16 V en la entrada del microcontrolador.
- La válvula de purgado de la PEMFC ( $C_4$ ) también trabaja con una señal todo/nada (1 corresponde a la válvula abierta y 0 corresponde a válvula cerrada), y debe abrirse cuando la tensión de celda de la pila sea inferior a la tensión de celda mínima de la pila ( $V_P < V_{P(min)}$ ).

La tensión de celda mínima de la pila ( $V_{P(min)}$ ) debe cumplir la siguiente expresión:

$$V_{p(min)} = \frac{V_S}{N^{\circ} \text{ celdas}} - 10\% \text{ (V)} \quad (\text{Ec.8})$$

Siendo  $V_S$  la tensión de salida de la pila, el número de celdas de la pila es de 20, y el 10% la diferencia entre  $V_P$  y  $V_{P(min)}$  permitida, para evitar una gran acumulación de impurezas en los canales de distribución del aire. El valor nominal de la tensión a plena carga de celda es de 0.6 V.

En el presente TFG la tensión de celda se ha entregado mediante una fuente de alimentación y un potenciómetro, en la que una tensión de celda de 1.25 V corresponde a una tensión de 2.7 V en la entrada del microcontrolador, y una tensión de celda de 0.2 V corresponde a una tensión de 0.43 V a la entrada del microcontrolador tras un acondicionamiento de la señal.

La válvula de purgado debe de cerrarse si el tiempo de apertura supera el tiempo de apertura máximo de 500 ms ( $T_{apertura} < T_{apertura (máx)}$ ).

- El control del convertidor DC/DC del motor del ventilador que incorpora el PEMFC real ( $C_2$ ) consiste en introducir una señal entre 0 y 3.3 V (donde 0 es el mínimo caudal de aire y 3 es el máximo caudal de aire) que depende de la temperatura de la pila ( $T^o$ ) y de la corriente de salida del PEMFC.

Por un lado, una temperatura inferior a 40 °C supone que el caudal va a ser proporcional a la corriente de salida del PEMFC ( $I$ ). Por otro lado, una vez la temperatura supera los 40 °C, el control debe poner la potencia al 50% (2.5 V), e irá aumentando proporcionalmente a la temperatura del PEMFC ( $T^o$ ) hasta alcanzar la máxima temperatura de la pila de 65 °C ( $T^o_{máx}$ ).

El caudal de aire también va a depender de la concentración de oxígeno ( $O_2$ ), donde un nivel inferior al 21% en el aire supone el aumento del caudal de aire para lograr completar la reacción en la pila de combustible.

La variación del caudal de aire ( $\Delta Q$ ) se calcula de la siguiente manera:

$$\Delta Q = \frac{0.21}{V_{SENSOR(O_2)}} (\% O_2 / V) \quad (Ec.8)$$

Donde 0.21 corresponde al 21% de oxígeno se encuentra en el aire atmosférico, y  $V_{SENSOR(O_2)}$  es la tensión de salida del sensor de oxígeno tras un acondicionamiento de señal, el cual entrega 2.7 V cuando la concentración es del 25%, y 2.05 V cuando la concentración es del 19%. En el presente TFG, este acondicionamiento se ha simulado y se entregado la señal mediante una fuente de alimentación.

Esta variación de caudal es la que determina que caudal de aire hay que introducir, multiplicando el valor de la variación por la tensión actual del convertidor del ventilador (0 a 3.3 V).

- El relé de seguridad del PEMFC ( $C_5$ ) trabaja con una señal todo/nada (1 corresponde a la válvula abierta y 0 corresponde a válvula cerrada), y se mantendrá cerrada mientras no ocurra alguno de los casos mencionados anteriormente para el cierre de la válvula principal ( $C_1$ ).
- El convertidor DC/DC de la salida del PEMFC ( $C_3$ ), además de interrumpir o proporcionar la corriente de carga de la batería, también se encarga de proporcionar la corriente necesaria de consumo del dron. Por lo que, si se cumple la condición de equilibrio de celdas de la ecuación estando en carga, el convertidor debe de entregar al menos 1 A adicional a la corriente de consumo del dron ( $C_3 = I_m + 1 A$ ).

Se ha decidido realizar una separación en tres máquinas de estados finitos (Purgado, Ventilador, y Convertidor de la salida), que se ejecutan al mismo tiempo y así cumplir con el determinismo. Sus diagramas de estados, flujogramas y codificación en C, se encuentran en el Anexo I.

Las tensiones que ha leído el microcontrolador se han acondicionado para que no superen los 2.7 V debido a las limitaciones del ADC, incluyendo las tensiones de la batería real, las tensiones del sensor de corriente de consumo del dron. Las tensiones provenientes de la pila de combustible simulada con una fuente de alimentación y potenciómetros se han entregado directamente con los valores que entregaría la pila de combustible real tras un acondicionamiento de señal adecuado.

Se han empleado dos sensores de corriente ACS712 como los que se muestran en la Figura 22 para la medición de la corriente de consumo del dron y la corriente de salida del PEMFC.

En el actual (TFG), se intentó emplear la comunicación RS232 para la adquisición de datos en el ordenador, y así poder trazar gráficas del comportamiento de las tensiones de celda y las corrientes de consumo del dron y de salida de la pila en MATLAB, pero no se logró establecer la comunicación con el programa, utilizando las siguientes líneas de código:

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

```
278 //Mandar datos al PC (ordenador)
279 mandar_datos_a_PC[0] = (uint8_t)((valor_adc[7]>>8) & 0x00FF); //V1(MSB)
280 mandar_datos_a_PC[1] = (uint8_t)(valor_adc[7] & 0x00FF); //V1(LSB)
281 mandar_datos_a_PC[2] = (uint8_t)((valor_adc[8]>>8) & 0x00FF); //V2(MSB)
282 mandar_datos_a_PC[3] = (uint8_t)(valor_adc[8] & 0x00FF); //V2(LSB)
283 mandar_datos_a_PC[4] = (uint8_t)((valor_adc[9]>>8) & 0x00FF); //V3(MSB)
284 mandar_datos_a_PC[5] = (uint8_t)(valor_adc[9] & 0x00FF); //V3(LSB)
285 mandar_datos_a_PC[6] = (uint8_t)((valor_adc[0]>>8) & 0x00FF); //V4(MSB)
286 mandar_datos_a_PC[7] = (uint8_t)(valor_adc[0] & 0x00FF); //V4(LSB)
287 mandar_datos_a_PC[8] = (uint8_t)((valor_adc[2]>>8) & 0x00FF); //I(MSB)
288 mandar_datos_a_PC[9] = (uint8_t)(valor_adc[2] & 0x00FF); //I(LSB)
289 mandar_datos_a_PC[10] = (uint8_t)((valor_adc[1]>>8) & 0x00FF); //Im(MSB)
290 mandar_datos_a_PC[11] = (uint8_t)(valor_adc[1] & 0x00FF); //Im(LSB)
291
292 //Mandar datos vía UART5 y UART7 con DMA
293 HAL_UART_Transmit_DMA(&huart5, mandar_datos_a_PC, sizeof(mandar_datos_a_PC));
294 HAL_UART_Transmit_DMA(&huart7, mandar_datos_a_PC, sizeof(mandar_datos_a_PC));
```

**Figura 26.** Comunicación UART para transmisión de datos entre el microcontrolador y el ordenador.

Y empleando el siguiente código en el programa MATLAB, el cual lee los valores y los almacena en un vector, para más tarde graficar los valores con la función ‘plot’:

```
25 %% Inicializar variables
26 t_muestrear = min*60 + seg; % Segundos a muestrear
27 muestras = t_muestrear * fs; % N muestras correspondiente al "tiempo a muestrear"
28 n = 0;
29
30 % Inicializacion de vectores para almacenar datos
31 t = -1*ones(muestras,1);
32 V1 = -1*ones(muestras,1);
33 V2 = -1*ones(muestras,1);
34 V3 = -1*ones(muestras,1);
35 V4 = -1*ones(muestras,1);
36 VBAT = -1*ones(muestras,1);
37 I = -1*ones(muestras,1);
38 Im = -1*ones(muestras,1);
39
40
41 %% Habilitar lectura de datos
42 writeline(serialportObj,"11");
43
44 %% Lectura de datos
45 while (n < muestras)
46 data_raw = read(serialportObj,14,"uint8");
47 t(n+1) = n * (1/fs);
48 V1(n+1) = (bitor(bitshift(data_raw(1,1), 8), data_raw(1,2))*Bit2V1);
49 V2(n+1) = (bitor(bitshift(data_raw(1,3), 8), data_raw(1,4))*Bit2V2);
50 V3(n+1) = (bitor(bitshift(data_raw(1,5), 8), data_raw(1,6))*Bit2V3);
51 V4(n+1) = (bitor(bitshift(data_raw(1,7), 8), data_raw(1,8))*Bit2V4);
52 VBAT(n+1) = (bitor(bitshift(data_raw(1,9), 8), data_raw(1,10))*Bit2VBAT);
53 I(n+1) = (bitor(bitshift(data_raw(1,11), 8), data_raw(1,12))*Bit2I);
54 Im(n+1) = (bitor(bitshift(data_raw(1,13), 8), data_raw(1,14))*Bit2Im);
55
```

**Figura 27.** Lectura de datos en MATLAB.

Por lo que, en el actual trabajo se ha diseñado e implementado únicamente en la placa de interfaz de hardware como parte de un desarrollo futuro del proyecto.

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

También se configuraron los periféricos necesarios como se muestra en la Figura 28, donde se observan configurados dos DAC y el ADC2 con DMA (Direct Memory Access), la UART5, la UART7, para una futura implementación en el software, y el Timer 8 para el disparo del ADC. Además, también se activaron siete entradas y seis salidas digitales. Estas configuraciones se muestran en el Anexo III.

```
47 /* Private variables -----*/
48 ADC_HandleTypeDef hadc2;
49 DMA_HandleTypeDef hdma_adc2;
50
51 DAC_HandleTypeDef hdac;
52 DMA_HandleTypeDef hdma_dac2;
53 DMA_HandleTypeDef hdma_dac1;
54
55 TIM_HandleTypeDef htim8;
56
57 UART_HandleTypeDef huart5;
58 UART_HandleTypeDef huart7;
59 DMA_HandleTypeDef hdma_uart5_rx;
60 DMA_HandleTypeDef hdma_uart5_tx;
61 DMA_HandleTypeDef hdma_uart7_rx;
62 DMA_HandleTypeDef hdma_uart7_tx;
```

Figura 28. Configuración de los periféricos.

Tras la correcta configuración y generación del código, se implementaron los estados, transiciones y eventos en C de la siguiente manera:

```
163 static fsm_trans_t table_H2DRONE1[] = {
164     { VALVULA_PURGADO, tension_de_celda_baja, PURGADO, abrir_C4 },
165     { PURGADO, tiempo_apertura_C4, VALVULA_PURGADO, cerrar_C4 },
166     { VALVULA_PURGADO, parametros_de_seguridad, PARO_DE_SEGURIDAD, abrir_circuito },
167     { PARO_DE_SEGURIDAD, parametros_rango_normal, VALVULA_PURGADO, cerrar_circuito },
168     { PURGADO, parametros_de_seguridad, PARO_DE_SEGURIDAD, abrir_circuito },
169     { PARO_DE_SEGURIDAD, parametros_rango_normal, VALVULA_PURGADO, cerrar_circuito },
170     { -1, NULL, -1, NULL },
171 };
172 //FSM PARA VENTILADOR DE LA PILA
173 static fsm_trans_t table_H2DRONE2[] = {
174     { VENTILADOR, temperatura_alta, VENTILADOR_A_ALTA_TEMPERATURA, ventilador_carga_media },
175     { VENTILADOR_A_ALTA_TEMPERATURA, temperatura_baja, VENTILADOR_A_BAJA_TEMPERATURA, ventilador_proporcional_I_Q },
176     { VENTILADOR_A_BAJA_TEMPERATURA, temperatura_alta, VENTILADOR_A_ALTA_TEMPERATURA, ventilador_carga_media },
177     { VENTILADOR_A_ALTA_TEMPERATURA, temperatura_alta, VENTILADOR, ventilador_proporcional_T_Q },
178     { VENTILADOR, parametros_de_seguridad, PARO_DE_SEGURIDAD, abrir_circuito },
179     { PARO_DE_SEGURIDAD, parametros_rango_normal, VENTILADOR, cerrar_circuito },
180     { VENTILADOR_A_ALTA_TEMPERATURA, parametros_de_seguridad, PARO_DE_SEGURIDAD, abrir_circuito },
181     { PARO_DE_SEGURIDAD, parametros_rango_normal, VENTILADOR, cerrar_circuito },
182     { VENTILADOR_A_BAJA_TEMPERATURA, parametros_de_seguridad, PARO_DE_SEGURIDAD, abrir_circuito },
183     { PARO_DE_SEGURIDAD, parametros_rango_normal, VENTILADOR, cerrar_circuito },
184     { -1, NULL, -1, NULL },
185 };
186 //FSM PARA CONVERTIDOR DC/DC DE LA SALIDA DE LA PILA
187 static fsm_trans_t table_H2DRONE3[] = {
188     { CONVERTIDOR_DCDC, bat_desequilibrada_or_max, INTERRUPCION_CARGA, corriente_de_consumo },
189     { INTERRUPCION_CARGA, bat_equilibrada, CONVERTIDOR_DCDC, carga_normal },
190     { CONVERTIDOR_DCDC, parametros_de_seguridad, PARO_DE_SEGURIDAD, abrir_circuito },
191     { PARO_DE_SEGURIDAD, parametros_rango_normal, CONVERTIDOR_DCDC, cerrar_circuito },
192     { INTERRUPCION_CARGA, parametros_de_seguridad, PARO_DE_SEGURIDAD, abrir_circuito },
193     { PARO_DE_SEGURIDAD, parametros_rango_normal, CONVERTIDOR_DCDC, cerrar_circuito },
194     { -1, NULL, -1, NULL },
195 };
196
```

Figura 29. Estados, transiciones y eventos de las FSM en C.

Donde cada fila está compuesta por: {ESTADO\_ACTUAL, función de guarda, ESTADO\_SIGUIENTE, función de acción}.

Las funciones de guarda devuelven un 1 si se ha cumplido la condición, y un 0 en caso contrario, es por ello por lo que están definidas como ‘uint8\_t’.



## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

Las funciones de acción son las encargadas de generar una señal de salida (digital o con el DAC) dependiente de las señales de entrada.

Las máquinas de estado se ejecutan periódicamente en el bucle principal del main() con un timer virtual (fue necesario incluir el fichero “vtimers.h”) configurado a 20 ms.

El disparo de las máquinas de estado se muestra en la Figura 30:

```
268 while (1)
269 {
270     //Ejecución de la FSM
271     if (timer_ended(&t_fire_fsm) == 1){
272
273         // Fire FSM
274         fsm_fire(fsm_H2DRONE1);
275         fsm_fire(fsm_H2DRONE2);
276         fsm_fire(fsm_H2DRONE3);
277
278         //Reiniciar Timer virtual para ejecutar FSMs, T_FIRE es la duracion del timer (20 ms)
279         timer_set(&t_fire_fsm,T_FIRE);
280
281     }
```

Figura 30. Disparo de las FSM.

Siendo ‘fsm\_H2DRONE1’ la máquina de estados de la válvula de purgado, ‘fsmH2DRONE2’ la del ventilador, y ‘fsm\_H2DRONE3’ la del convertidor DC/DC de la salida de la pila.

La inicialización de las funciones de guarda y de acción son las siguientes:

```
129/* ----- Guard Functions ----- */
130 // H2DRONE -> Funciones de las condiciones (guardas)
131 static uint8_t bat_desequilibrada_or_max (fsm_t* this);
132 static uint8_t bat_equilibrada (fsm_t* this);
133 static uint8_t temperatura_alta (fsm_t* this);
134 static uint8_t temperatura_baja (fsm_t* this);
135 static uint8_t tension_de_celda_baja (fsm_t* this);
136 static uint8_t tiempo_apertura_C4 (fsm_t* this);
137 static uint8_t parametros_de_seguridad (fsm_t* this);
138 static uint8_t parametros_rango_normal (fsm_t* this);
139
140
141/* ----- Action Functions ----- */
142 // H2DRONE -> Funciones de las acciones
143 static void corriente_de_consumo (fsm_t* this);
144 static void carga_normal (fsm_t* this);
145 static void ventilador_carga_media (fsm_t* this);
146 static void ventilador_proporcional_I_Q (fsm_t* this);
147 static void ventilador_prporcional_T_Q (fsm_t* this);
148 static void abrir_C4 (fsm_t* this);
149 static void cerrar_C4(fsm_t* this);
150 static void abrir_circuito(fsm_t* this);
151 static void cerrar_circuito(fsm_t* this);
```

Figura 31. Declaración de las funciones de guarda y acción.

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

Como se ha mencionado anteriormente, para el disparo del ADC2, se ha empleado un Timer, ya que permite realizar el disparo de forma periódica. Este Timer ha sido configurado para disparar cada 20 ms, al igual que las máquinas de estados.

En la función de “Callback” (se llama a esta función después de finalizar cada conversión) se han realizado las conversiones a su correspondiente unidad de los 11 valores del ADC2 leídos en bits, como se observa en la Figura 32:

```
867=void HAL_ADC_ConvPpltCallback(ADC_HandleTypeDef* hadc)
868 {
869     //Conversión de bits a Voltios reales (V)
870     volts[0]= (((float)valor_adc[0] / RESOLUCION) * MAX_VOLTS_V4)*GANANCIA_V4; //V4
871     volts[1]= (((float)valor_adc[1] / RESOLUCION) * MAX_VOLTS_DRON; //SENSOR DE CORRIENTE DRON
872     volts[2]= (((float)valor_adc[2] / RESOLUCION) * MAX_VOLTS_PILA; //SENSOR DE CORRIENTE PILA
873     volts[3]= (((float)valor_adc[3] / RESOLUCION) * MAX_VOLTS_ADC; //Presión (P)
874     volts[4]= (((float)valor_adc[4] / RESOLUCION) * MAX_VOLTS_ADC)*GANANCIA_VS; //Tensión de salida de la pila (Vs)
875     volts[5]= (((float)valor_adc[5] / RESOLUCION) * MAX_VOLTS_ADC; //Temperatura de la pila (T)
876     volts[6]= (((float)valor_adc[6] / RESOLUCION) * MAX_VOLTS_ADC)*GANANCIA_VP; //Tensión de celda (Vp)
877     volts[7]= (((float)valor_adc[7] / RESOLUCION) * MAX_VOLTS_V1)*GANANCIA_V1; //V1
878     volts[8]= (((float)valor_adc[8] / RESOLUCION) * MAX_VOLTS_V2)*GANANCIA_V2; //V2
879     volts[9]= (((float)valor_adc[9] / RESOLUCION) * MAX_VOLTS_V3)*GANANCIA_V3; //V3
880     volts[10]= (((float)valor_adc[10] / RESOLUCION) * MAX_VOLTS_ADC; //Flujo de oxígeno (O2)
881
882     // Las señales de los sensores de corriente se cambian de nombre y se pasan a Amperios reales (2.7V -> 3 A)
883     corriente[0]=volts[1]*GANANCIA_VDRON; // Ii
884     corriente[1]=volts[2]*GANANCIA_VPILA; // I
885     // Las señal del nivel de presión se cambia de nombre y se pasa a unidades de presión (2.7 V -> 1 bar)
886     presion=volts[3]*GANANCIA_PRESION; // P
887     // Las señal de la concentración de oxígeno se cambia de nombre y se pasa a unidades de concentración (2.7 V -> 21 % O2)
888     concentracion_O2=volts[10]*GANANCIA_O2; // O2
889     // Las señal de la temperatura se cambia de nombre y se pasa a unidades de temperatura (2.7 V -> 70 °C)
890     temperatura=volts[5]*GANANCIA_T; // T
891     //Actualización de las variables antes de ejecutar la FSM
892     //Indicadores de terminación del timer 1: si se se ha acabado, 0: si no se ha acabado
893     flag_tiempo_C4=timer_ended(&timer_C4);
894     //Indicador visual de conversión finalizada (LED3 ON/LED3 OFF)
895     HAL_GPIO_TogglePin(GPIO6, LD3_Pin);
896 }
897
```

Figura 32. Conversión de datos tras la lectura del ADC2.

A partir de los valores leídos en su unidad correspondiente, el control comprueba que función de guarda se cumple, para realizar la llamada a la función de acción.

En la siguiente Figura 33 se muestran dos ejemplos de funciones de guarda, concretamente las funciones que indican si la temperatura es alta ( $T > 40$ ) o si es baja ( $T < 40$ ).

```
960=static uint8_t temperatura_alta (fsm_t* this){
961     //Se activa cuando la temperatura es igual o supera los 40 °C hasta los 65 °C
962     if((TEMP_MAX >= temperatura) && (temperatura >= TEMP_MIN)){
963         return(1);
964     }
965     }
966     else{
967         return(0);
968     }
969 }
970 }
971=static uint8_t temperatura_baja (fsm_t* this){
972     //Se activa cuando la temperatura es inferior a 40 °C
973     if(temperatura<TEMP_MIN){
974         return(1);
975     }
976     }
977     else{
978         return(0);
979     }
980 }
981 }
```

Figura 33. Ejemplos funciones de guarda.

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

Un ejemplo de función de acción se muestra en la Figura 34, concretamente la función de apertura y cierre de la válvula de purgado:

```
1120 static void abrir_C4 (fsm_t* this){
1121     HAL_GPIO_WritePin(GPIOF, GPIO_PIN_10, 1); //Abrir válvula de purgado (C4)
1122     timer_set(&timer_C4, TIEMPO_APERTURA_C4); //Set timer virtual para abrir válvula C4 durante 500 ms.
1123 }
1124 }
1125 static void cerrar_C4 (fsm_t* this){
1126     HAL_GPIO_WritePin(GPIOF, GPIO_PIN_10, 0); //Cerrar válvula de purgado (C4)
1127     timer_disable(&timer_C4); //Deshabilitar timer virtual
1128 }
1129 }
```

**Figura 34.** Ejemplos funciones de acción.

Como se observa, activa o desactiva la salida digital del pin 10, que corresponde a la válvula de purgado, la cual una vez abierta debe cerrarse a los 500 ms. Para temporizar dicho tiempo de apertura, se empleó un Timer virtual.

Otro ejemplo de función de acción se muestra en la Figura 35:

```
1047 static void corriente_de_consumo (fsm_t* this){
1048     // Convertir corriente a tensión proporcional (3.0 es la máxima tensión del DAC)
1049     volatile float volts_corriente_consumo = (corriente[0]/I_MAX)*VDAC_MAX;
1050 }
1051 // Convertir el valor de tensión a la escala del DAC (0 a 4095)
1052 dac_value2= (uint32_t)(volts_corriente_consumo*DIV_RESOLUCION_VDACMAX);
1053 }
1054 }
```

**Figura 35.** Ejemplo función de acción sobre el DAC2.

Esta función interrumpe la corriente de carga igualándola a la corriente de consumo. Como se observa, las funciones de acción que actúan sobre alguno de los DAC se encargan de proporcionarle el valor correcto en bits a la variable de almacenamiento, que en este caso es “dac\_value2”.

Cabe destacar que la gran mayoría de parámetros se han definido como constantes en el fichero ‘main.h’, con #defines para permitir una fácil modificación de los valores si es necesario, como se muestra en la Figura 36:

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

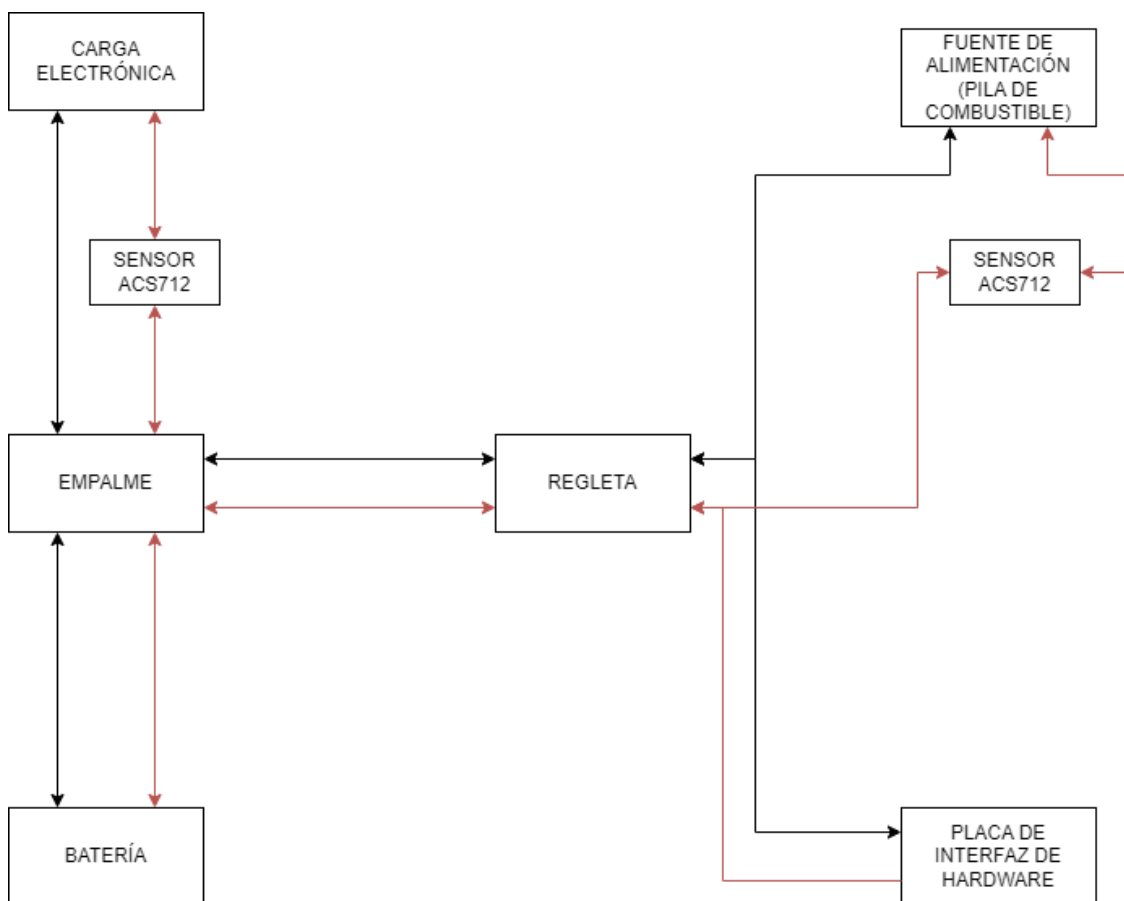
```
129 //CORRIENTE MÁXIMA DE SALIDA DE LA PILA
130 #define I_MAX 3.00
131
132 //TEMPERATURA DE LA PILA MÍNIMA Y MÁXIMA
133 #define TEMP_MAX 65.0
134 #define TEMP_MIN 40.0
135
136 //PRESIÓN MÁXIMA DEL DEPÓSITO DE H2 DE LA PILA
137 #define PRESION_MAX 0.6
138
139 //TENSION DE SALIDA DE LA PILA MÍNIMA Y MÁXIMA
140 #define VS_MAX 24.0
141 #define VS_MIN 10.0
142
143
144 //TENSIONES MÁXIMAS DE ENTRADA AL ADC
145 #define MAX_VOLTS_ADC 2.70
146 #define MAX_VOLTS_V1 2.707
147 #define MAX_VOLTS_V2 2.665
148 #define MAX_VOLTS_V3 2.702
149 #define MAX_VOLTS_V4 2.697
150 #define MAX_VOLTS_DRON 2.866
151 #define MAX_VOLTS_PILA 2.838
```

**Figura 36.** Ejemplo de definiciones de constantes en el main.h.

## 5. RESULTADOS Y DISCUSIÓN

Tras la correcta implementación de las máquinas de estado en C, se realizó la validación del funcionamiento del sistema. Como se ha mencionado anteriormente, las señales provenientes de la pila de combustible se simularon con una fuente de alimentación, y el dron con una carga electrónica.

La validación consistió en ejecutar el control, realizando las conexiones del sistema que se muestran en la Figura 35 y realizar mediciones de las salidas digitales y analógicas con un multímetro. El procedimiento del conexionado y las especificaciones de los cables empleados se encuentran en el Anexo II.



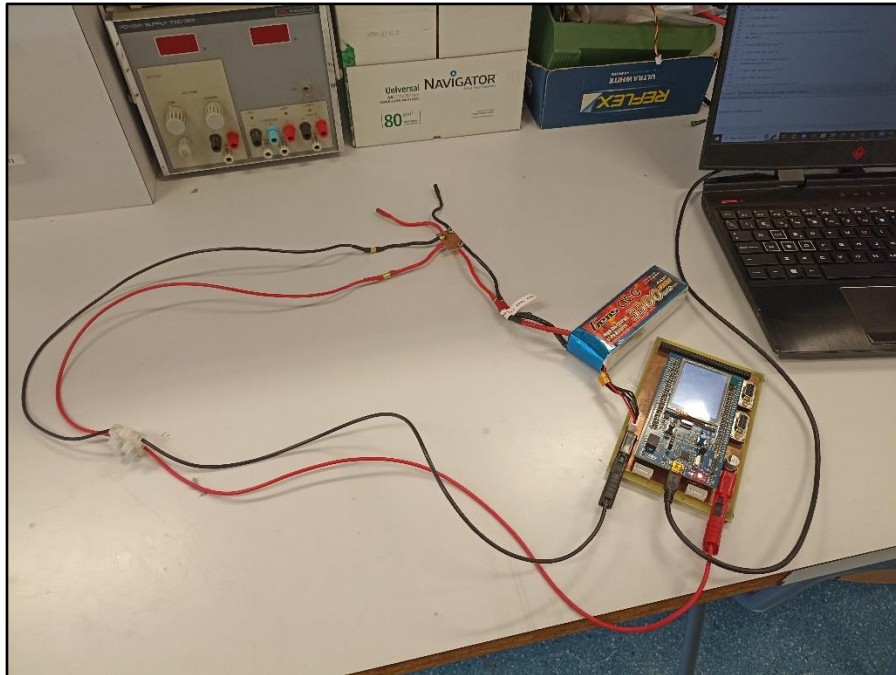
**Figura 37.** Esquema de las conexiones para la validación del sistema.

Donde las flechas rojas indican que son cables conectados al terminal positivo y las flechas negras indican que son cables conectados al terminal negativo.

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

En primer lugar, se realizó la validación de la lectura de las tensiones de la batería ( $V_1, V_2, V_3, V_4, V_{BAT}$ ) en el modo de depuración del STM32CubeIDE, conectado la batería a la placa de interfaz de hardware, como se muestra en la Figura 36:



**Figura 38.** Validación de lectura de tensiones de batería.

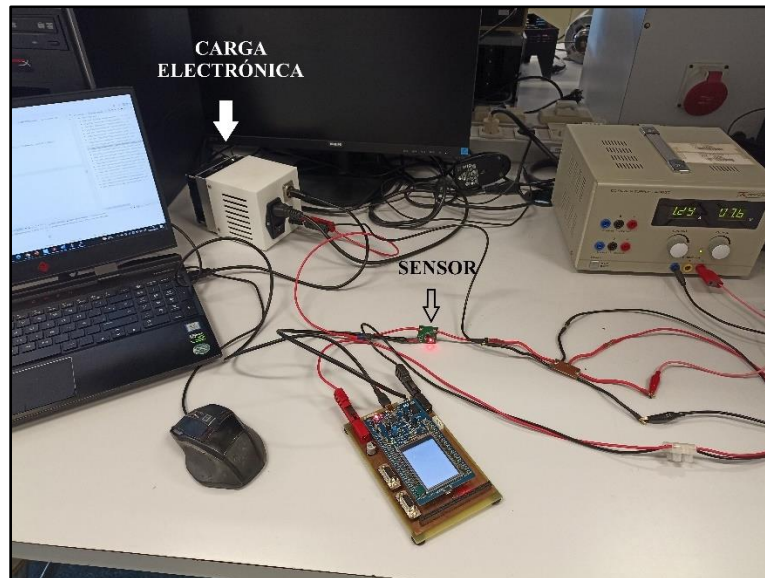
Leyendo unas tensiones de celda que se muestran en la Figura 37:

(*)= V1	float	3.14857888
(*)= V2	float	7.71142054
(*)= V3	float	12.3594131
(*)= V4	float	15.6743116
(*)= VBAT	float	15.6743116

**Figura 39.** Lectura de las tensiones de celda de la batería.

Tras una primera validación de la lectura de las tensiones de la batería, se realizó la comprobación de la lectura de la señal entregada por el sensor de corriente de consumo del dron, conectando el sensor a la carga electrónica y a la placa de interfaz de hardware, como se muestra en la Figura 38:

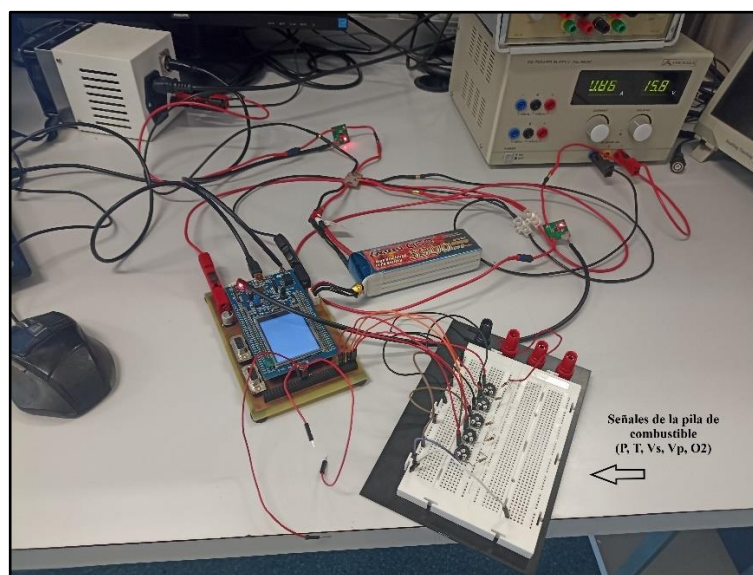
## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”



**Figura 40.** Validación lectura de corriente con la carga electrónica y el sensor de corriente.

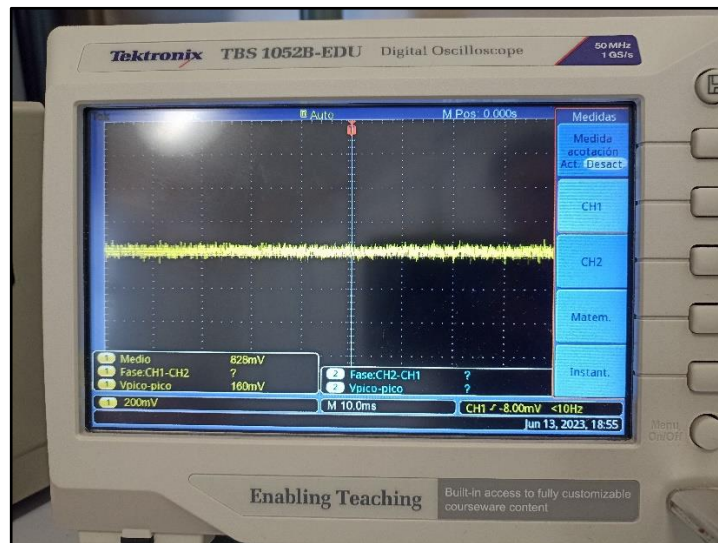
El valor entregado por este sensor fue correcto, por lo que, a continuación, se comprobó la señal entregada por el sensor de corriente de salida de la pila (fuente de alimentación), conectando el sensor a la fuente de alimentación y a la placa de interfaz de hardware.

Una vez comprobadas las lecturas, se validó el funcionamiento del sistema completo, realizando las conexiones que se muestran en la Figura 35. El conexionado realizado se muestra en la siguiente Figura:



**Figura 41.** Conexionado del sistema completo.

Tras la validación del sistema en su totalidad, se comprobó que la lectura de los canales del ADC era correcta, aunque muy inestable. Para detectar el fallo del sistema, se midió alguno de los canales con un osciloscopio del laboratorio del DIE en la UPV, detectando una alta cantidad de ruido electromagnético, como se muestra en la Figura 40:



**Figura 42.** Medición con el osciloscopio de uno de los canales ADC.

Este error se intentó prevenir mediante la conexión de filtros paso bajo en la entrada de los canales del ADC, pero algunas de las mejoras para la eliminación del ruido habrían sido las siguientes:

- Conexión de un amplificador operacional (*buffer*) en la entrada de los canales del ADC. Este amplificador operacional actúa como barrera, evitando que la carga del ADC afecta a la señal de entrada y estabilizando y manteniendo constante el voltaje de la señal de entrada.

El sistema cuenta con 12 canales ADC, por lo que tres unidades del circuito integrado MCP6024 servirían para mejorar la calidad de todas las señales. Este circuito integrado cuenta con cuatro amplificadores operacionales, y con una entrada “rail-to-rail” [33], por lo que no habría problema con alimentarlo de 0 a 5 V mediante la señal del regulador 7805.



## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

- Realizando una buena separación de masas (analógica y digital), reduciendo así las interferencias, evitando la contaminación de referencias, minimizando la distorsión y preservando la calidad de la señal.
- Uniendo las dos masas en un pequeño y estrecho punto, que favorece la eliminación de las diferencias de potencial existentes entre las dos masas, establecer una referencia común, y suprimir los bucles de tierra.

Estas mejoras se implementarán en un futuro desarrollo del proyecto, para obtener una lectura limpia y estable.

Para una validación precisa del funcionamiento de las máquinas de estado, se empleó el modo depuración del STM32CubeIDE, utilizando “*breakpoints*” para pausar el programa e introducir los valores de entradas digitales y analógicas deseados.

En primer lugar, en la Figura 41 se muestra cómo actúan las salidas digitales del relé de seguridad y la válvula principal al haber superado la presión su valor máximo de seguridad de 0.6 bar.

(*)= V1	float	4
(*)= V2	float	8
(*)= V3	float	12
(*)= V4	float	16
(*)= Vs	float	20
(*)= Vp	float	1
(*)= presion	float	0.800000012
(*)= concentracion_O2	float	0.209999993
(*)= temperatura	float	50.5
(*)= estado_C1	uint8_t	0 '\0'
(*)= estado_C4	uint8_t	0 '\0'
(*)= estado_C5	uint8_t	1 '\001'
(*)= VBAT	float	16

**Figura 43.** Validación del relé de seguridad y la válvula principal en el modo depuración.

Como se observa, todos los parámetros se encuentran en su estado normal, a excepción de la presión, cuyo valor es 0.8 bar, causando así la apertura del relé de seguridad ( $C_5$ ) y el cierre de la válvula principal ( $C_1$ ).

También se comprobó el funcionamiento de los canales DAC, y la válvula de purgado ( $C_4$ ), como se muestra en la Figura 42:

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

corriente[0]	float	2
corriente[1]	float	2.5
V1	float	3.5999999
V2	float	7.5
V3	float	11.3999996
V4	float	15.8000002
Vs	float	21.1898823
Vp	float	0.960186839
presion	float	0.400000006
concentracion_O2	float	0.209999993
temperatura	float	46
estado_C1	uint8_t	1 '\001'
estado_C4	uint8_t	1 '\001'
estado_C5	uint8_t	0 '\0'
VBAT	float	15.8000002
dac_value1	uint32_t	3753
dac_value2	uint32_t	2730

**Figura 44.** Validación del funcionamiento de los canales DAC y la válvula de purgado.

Siendo *corriente [0]* la corriente que consume el dron ( $I_m$ ) y *corriente [1]* la corriente de salida de la pila, *dac\_value1* el valor en bits que se aplica en  $C_2$  y *dac\_value2* el valor en bits que se aplica en  $C_3$ .

Como se observa, los valores del DAC son los correspondientes, y la válvula de purgado ( $C_4$ ) está abierta, debido a que la tensión de celda de la pila ( $V_p$ ) es inferior a la tensión de celda mínima de la pila ( $V_{p(min)}$ ), que se calcula dividiendo la tensión de la pila ( $V_s$ ) entre el número de celdas de la pila (20), y restándole un 10%.

El correcto funcionamiento de estas máquinas de estado servirá en un futuro desarrollo del proyecto para controlar los dispositivos de una pila de combustible real, aplicando las mejoras para la eliminación de las interferencias electromagnéticas.

## 6. BIBLIOGRAFÍA

- [1] R. Marqués Gimeno, “Análisis y diseño de un sistema híbrido pila de combustible batería para alimentación de vehículos aéreos no tripulados,” Universitat Politècnica de València, 2022. Accessed: Feb. 16, 2023. [Online]. Available: <https://riunet.upv.es/handle/10251/180729>
- [2] “CO2 (Dióxido de carbono) | PRTR España.” <https://prtr-es.es/co2-dioxido-de-carbono,15590,11,2007.html> (accessed May 04, 2023).
- [3] G. Alsina, “Microcontrolador de seguridad para el sector automotriz -.” Accessed: Feb. 21, 2023. [Online]. Available: <https://www.seguridadprofesionalhoy.com/microcontrolador-de-seguridad-para-el-sector-automotriz/>
- [4] “Mavic Mini - DJI.” <https://www.dji.com/es/mavic-mini> (accessed May 15, 2023).
- [5] O. Almarza, “Drones híbridos con autonomía extendida a tener en cuenta”, Accessed: Mar. 21, 2023. [Online]. Available: <https://www.mundodeportivo.com/urbantecno/tutoriales/opciones-hibridas-autonomia-extendida-drones>
- [6] “Hybrix long-endurance drone – Quaternium.” <https://www.quaternium.com/uav/hybrix-drone/> (accessed May 15, 2023).
- [7] “H2 Dron Energy – Tecnología de hidrógeno para UAVs.” <http://h2dronenergy.com/> (accessed May 15, 2023).
- [8] “TRA BUSSO GROUP es parte de Programa Formativo de Hidrógeno Verde - Busso Group.” <https://www.bussogroup.com/2021/12/13/tra-busso-group-es-parte-de-programa-formativo-de-hidrogeno-verde/> (accessed May 15, 2023).
- [9] “Hidrógeno - ARIEMA”, Accessed: Feb. 23, 2023. [Online]. Available: <https://www.ariema.com/informacion-sobre-hidrogeno>
- [10] “Pilas de Combustible - Centro Nacional de Hidrógeno.” <https://www.cnh2.es/pilas-de-combustible/> (accessed Mar. 21, 2023).
- [11] “Pila de combustible - Wikipedia, la enciclopedia libre.” Accessed: Mar. 21, 2023. [Online]. Available: [https://es.wikipedia.org/wiki/Pila\\_de\\_combustible](https://es.wikipedia.org/wiki/Pila_de_combustible)
- [12] “Tipos de pilas de combustible y su clasificación | Apilados.” <https://apilados.com/blog/tipos-de-pilas-de-combustible-su-clasificacion/> (accessed May 15, 2023).

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

- [13] “Proceso de carga-descarga de una batería de iones de litio – Un poco de ciencia, por favor.” <https://blogs.publico.es/ignacio-martil/2019/11/28/proceso-de-carga-descarga-de-una-bateria-de-iones-de-litio/> (accessed May 05, 2023).
- [14] J. Jürgens, “El NMC es el material preferible para el cátodo de las baterías a iones de litio - LG Battery Blog Europe,” 2022. <https://lghomebatteryblog.eu/es/nmc-lfp-baterias-iones-litio/> (accessed Apr. 04, 2023).
- [15] I. Mártil, “Proceso de carga-descarga de una batería de iones de litio – Un poco de ciencia, por favor,” Nov. 28, 2019. Accessed: Mar. 24, 2023. [Online]. Available: <https://blogs.publico.es/ignacio-martil/2019/11/28/proceso-de-carga-descarga-de-una-bateria-de-iones-de-litio/>
- [16] G. Portillo, “Tipos de baterías: características, usos y diferencias con las pilas | Renovables Verdes.” <https://www.renovablesverdes.com/tipos-de-baterias/> (accessed Apr. 27, 2023).
- [17] “Lithium-Ion vs Lithium Polymer Battery - Robocraze.” <https://robocraze.com/blogs/post/lithium-ion-vs-lithium-polymer-battery> (accessed Apr. 27, 2023).
- [18] “Microcontrolador - Wikipedia, la enciclopedia libre.” <https://es.wikipedia.org/wiki/Microcontrolador> (accessed Mar. 27, 2023).
- [19] S. Paul Aguayo, “INTRODUCCIÓN AL MICROCONTROLADOR,” Nov. 2004, Accessed: Mar. 27, 2023. [Online]. Available: <http://usuarios.lycos.es/sfriswolkker>
- [20] R. Omrani, H. Q. Nguyen, and B. Shabani, “Open-cathode PEMFC heat utilisation to enhance hydrogen supply rate of metal hydride canisters,” *Energy Procedia*, vol. 160, pp. 542–549, Feb. 2019, doi: 10.1016/J.EGYPRO.2019.02.204.
- [21] “STMicroelectronics: Our technology starts with you.” [https://www.st.com/content/st\\_com/en.html](https://www.st.com/content/st_com/en.html) (accessed May 15, 2023).
- [22] O. Weis, “Última revisión sobre el protocolo RS232,” Apr. 02, 2020. <https://www.serial-port-monitor.org/es/articles/serial-communication/rs232-interface/> (accessed Apr. 20, 2023).
- [23] “H 100 | Profesional | Stacks a hidrógeno | H 100-300 | H2Planet - Re-evolution started - Hydrogen & fuel-cell experience.” <https://www.h2planet.eu/es/detail/H100> (accessed May 25, 2023).
- [24] J. Muñoz Pérez, “Monitorización del consumo de un dron cuadracóptero,” Oct. 2019, Accessed: Apr. 24, 2023. [Online]. Available: <https://riunet.upv.es:443/handle/10251/126972>

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

- [25] Newton C. Braga, “Lo que significa Rail-to-rail (RRO) (ART279S),” 2017. Accessed: Apr. 24, 2023. [Online]. Available: <http://www.incb.com.mx/index.php/articulos/12-informacion-y-ideas-practicas-ip000s/1679-lo-que-significa-rail-to-rail-rro-art279s>
- [26] “Comunicación RS-232 - Controles de movimiento ADVANCED.” <https://www.a-m-c.com/es/experiencia/technologies/peripheral-interface/rs-232/> (accessed Apr. 24, 2023).
- [27] “Qué es UART | Rohde & Schwarz.” [https://www.rohde-schwarz.com/es/productos/test-y-medida/essentials-test-equipment/digital-oscilloscopes/que-es-uart\\_254524.html](https://www.rohde-schwarz.com/es/productos/test-y-medida/essentials-test-equipment/digital-oscilloscopes/que-es-uart_254524.html) (accessed May 05, 2023).
- [28] S. Pithadia, “Grounding in mixed-signal systems demystified, Part 1.” [https://www.ti.com/lit/an/slyt499/slyt499.pdf?ts=1682363350794&ref\\_url=https%253A%252F%252Fwww.google.com%252F](https://www.ti.com/lit/an/slyt499/slyt499.pdf?ts=1682363350794&ref_url=https%253A%252F%252Fwww.google.com%252F) (accessed Apr. 25, 2023).
- [29] “Procesado Digital.” [https://www.ehu.eus/procesadoinsvirtual/T1\\_4para%20que%20necesitamos%20filtros%20antialiasing.html](https://www.ehu.eus/procesadoinsvirtual/T1_4para%20que%20necesitamos%20filtros%20antialiasing.html) (accessed Apr. 24, 2023).
- [30] “Filtro RC Paso Bajo - Electrónica Unicrom.” <https://unicrom.com/filtro-rc-paso-bajo/> (accessed Apr. 24, 2023).
- [31] F. Gimeno, “MAQUINAS DE ESTADOS FINITAS (MEF).” [https://poliformat.upv.es/access/content/group/GRA\\_12158\\_2022/TEORIA\\_SDA/T14\\_Sistemas\\_MEF\\_2833x.pdf](https://poliformat.upv.es/access/content/group/GRA_12158_2022/TEORIA_SDA/T14_Sistemas_MEF_2833x.pdf) (accessed Apr. 09, 2023).
- [32] E. Ashford Lee and S. Arankumar Seshia, *INTRODUCTION TO EMBEDDED SYSTEMS*, 2nd ed. 2017. Accessed: Jun. 10, 2023. [Online]. Available: [https://ptolemy.berkeley.edu/books/leeseshia/releases/LeeSeshia\\_DigitalV2\\_2.pdf](https://ptolemy.berkeley.edu/books/leeseshia/releases/LeeSeshia_DigitalV2_2.pdf)
- [33] “MCP6021/1R/2/3/4 Features,” 2009.





UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

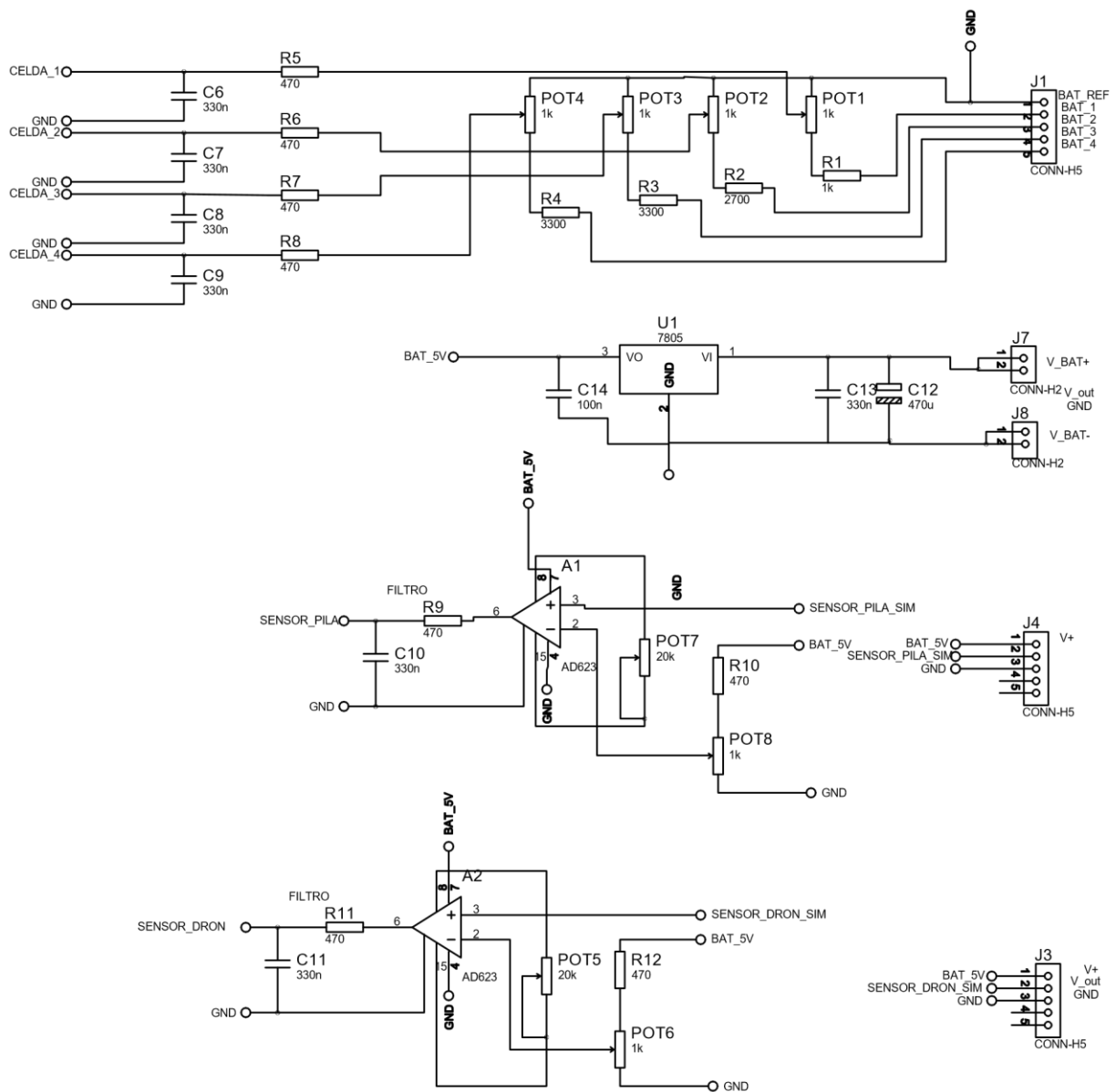


# PLANOS

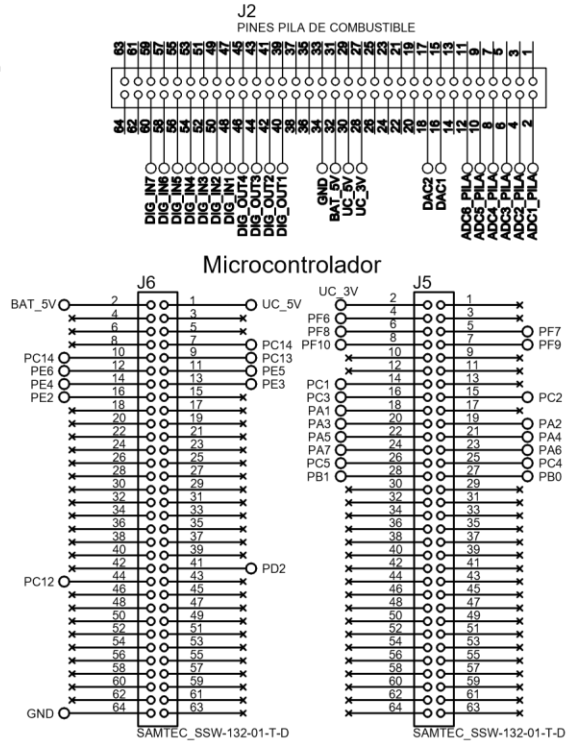
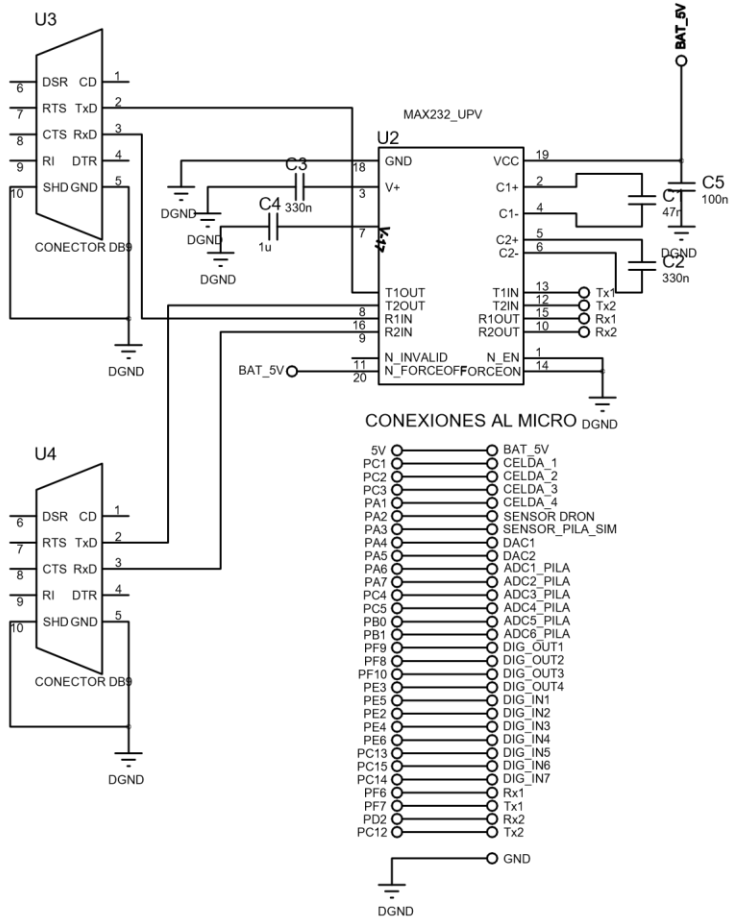
**Implementación de Máquina de  
Estados Finita en Microcontrolador  
para optimizar el control del sistema  
de energía de un dron híbrido “pila  
de combustible – batería”**



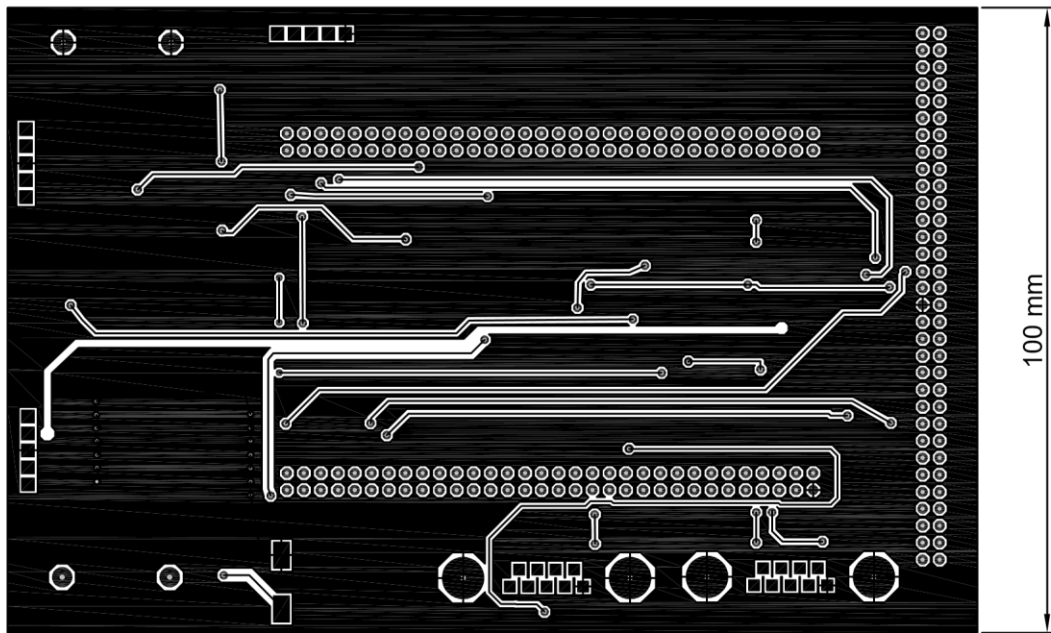




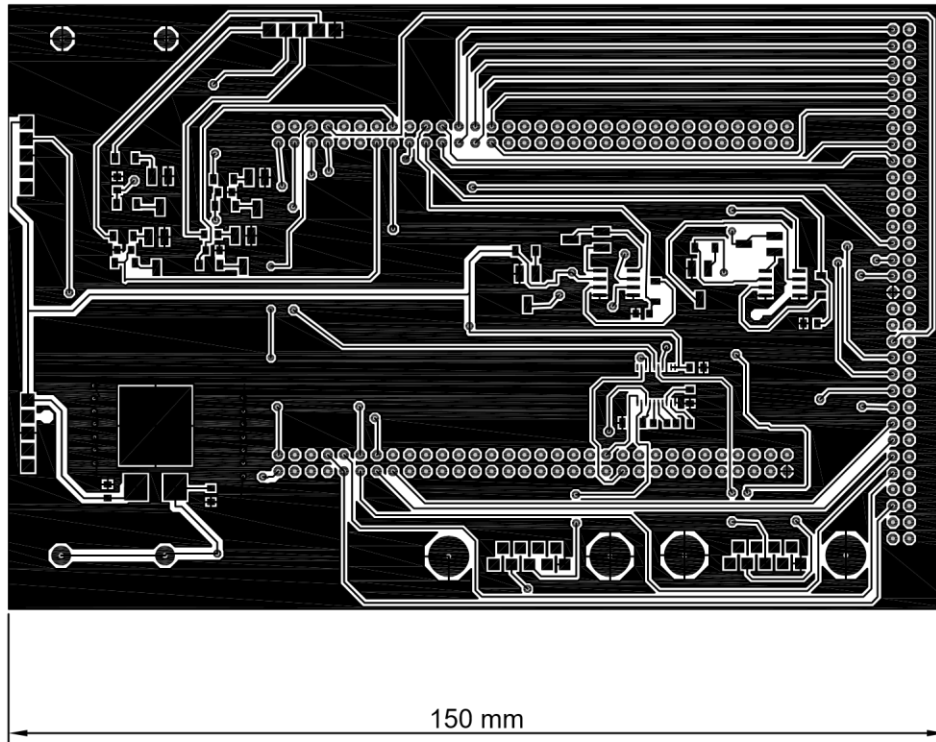
	Fecha	Nombre	Firma:	ESCUELA TÉCNICA SUPERIOR DE INGENIERIA DEL DISEÑO
Dibujado	19/05/2023	Walid El Bachouti Khalil		Curso: 4º
Comprobado				
id. s. normas				
Escala:	1:1		Descripción:	
Lamina:	Nº1		Esquema eléctrico en Proteus del acondicionamiento de señal de la batería y sensores de corriente	



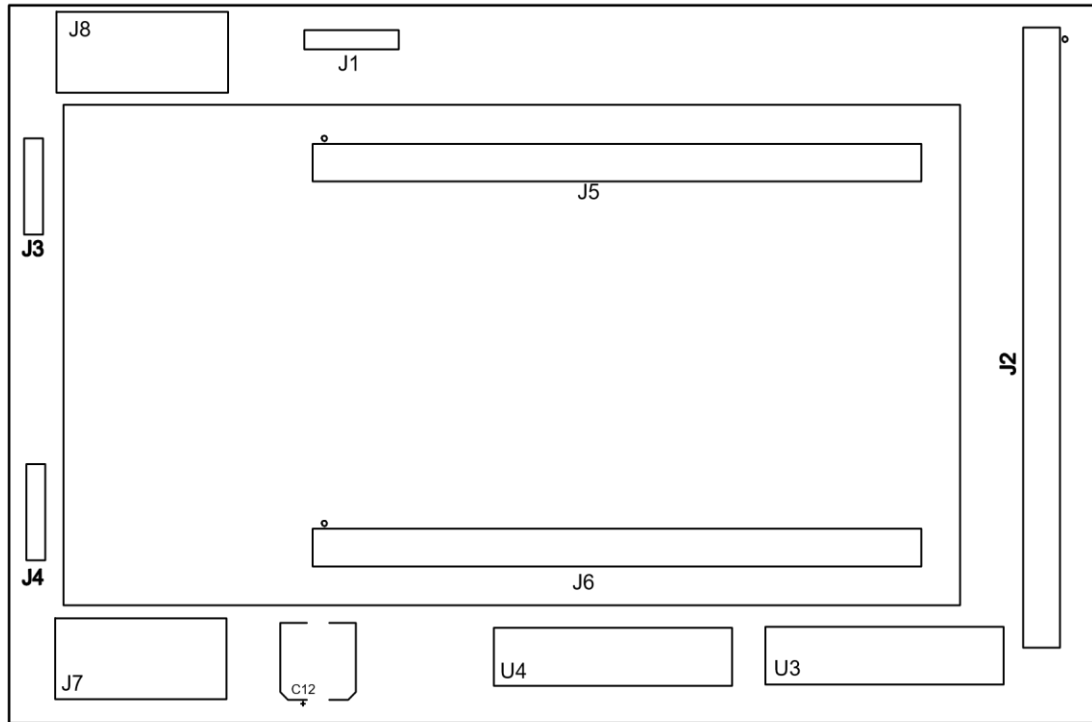
	Fecha	Nombre	Firma:	ESCUELA TÉCNICA SUPERIOR DE INGENIERIA DEL DISEÑO
Dibujado	19/05/2023	Walid El Bachouti Khalil		Curso: 4°
Comprobado				
id. s. normas				
Escala:	1:1		Descripción:	
Lamina:	Nº 2		Esquema eléctrico en Proteus del MAX-232 y el microcontrolador	



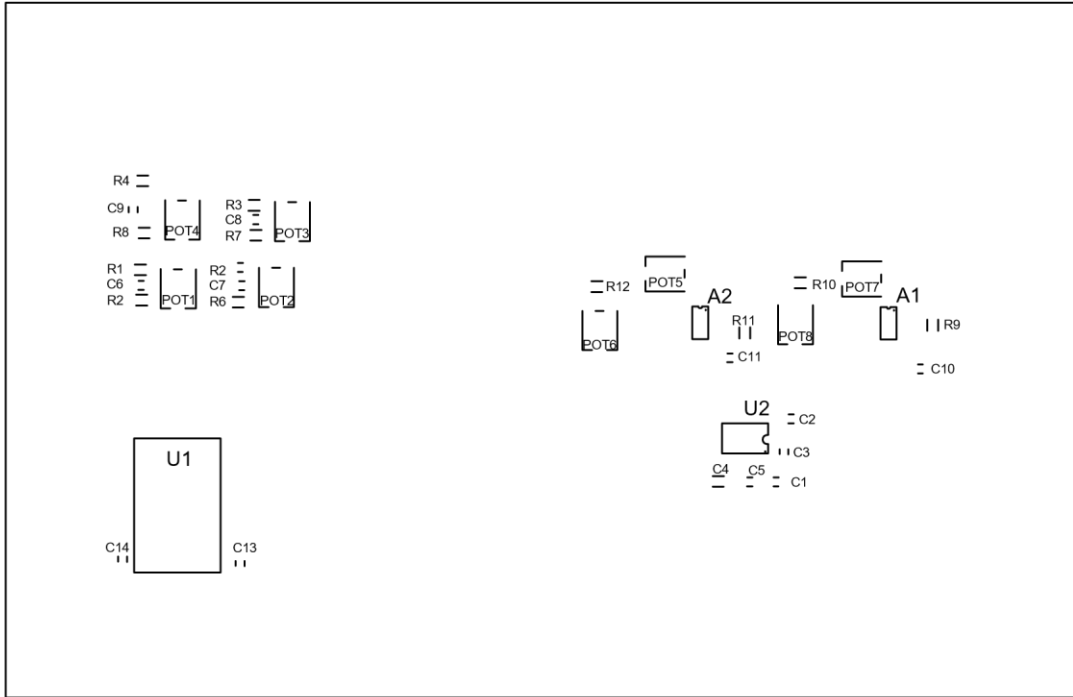
		<i>Fecha</i>	<i>Nombre</i>	<i>Firma:</i>	<i>ESCUELA TÉCNICA SUPERIOR DE INGENIERIA DEL DISEÑO Curso: 4º</i>
	<i>Dibujado</i>	<i>23/05/2023</i>	<i>Walid El Bachouti Khalil</i>		
	<i>Comprobado</i>				
	<i>id. s. normas</i>				
	<i>Escala:</i>	<i>1:1</i>		<i>Descripción:</i>	<i>Top Copper</i>
	<i>Lamina:</i>	<i>Nº 3</i>			



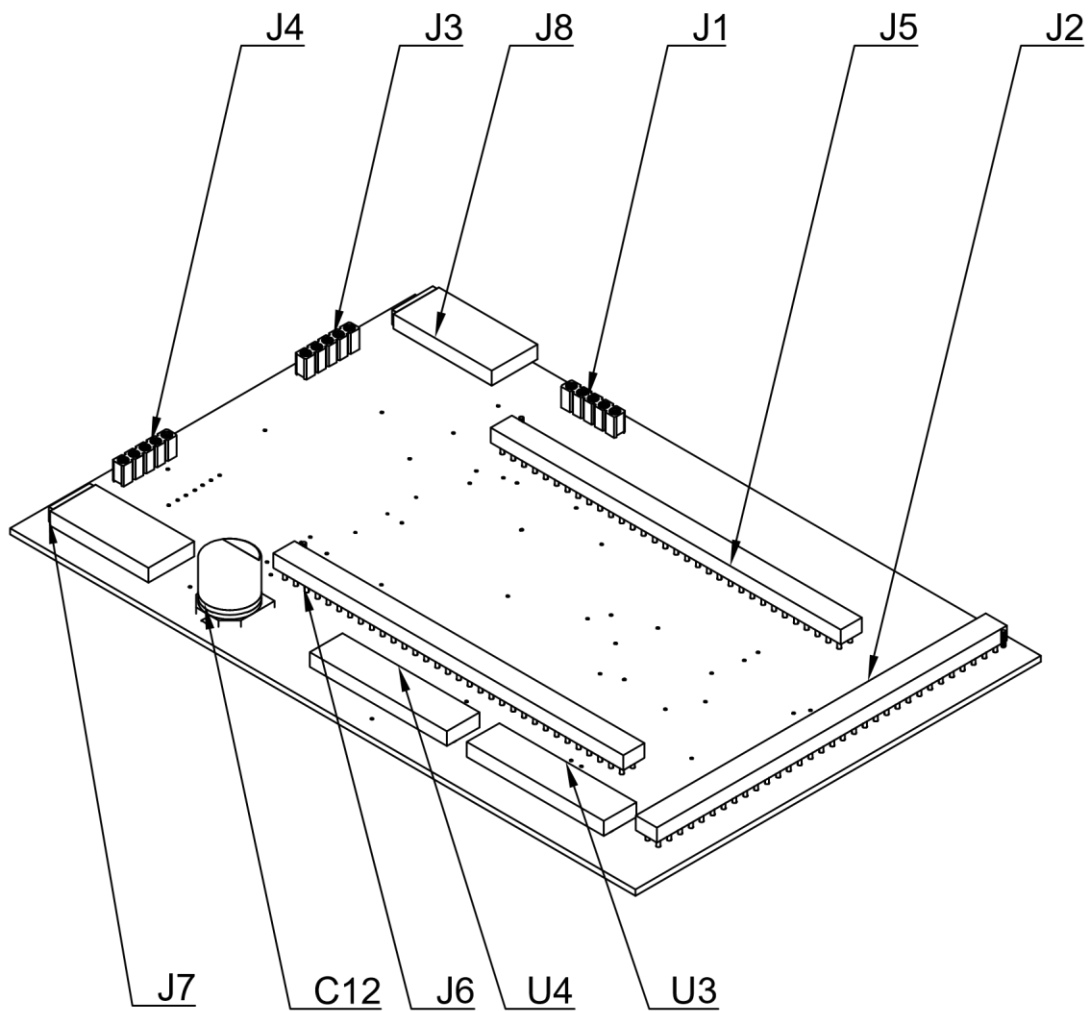
	<i>Fecha</i>	<i>Nombre</i>	<i>Firma:</i>	<i>ESCUELA TÉCNICA SUPERIOR DE INGENIERIA DEL DISEÑO</i>  <i>Curso: 4º</i>
<i>Dibujado</i>	<i>19/05/2023</i>	<i>Walid El Bachouti Khalil</i>		
<i>Comprobado</i>				
<i>id. s. normas</i>				
<i>Escala:</i>	<i>1:1</i>		<i>Descripción:</i>	
<i>Lamina:</i>	<i>Nº 4</i>		<i>Bottom Copper</i>	



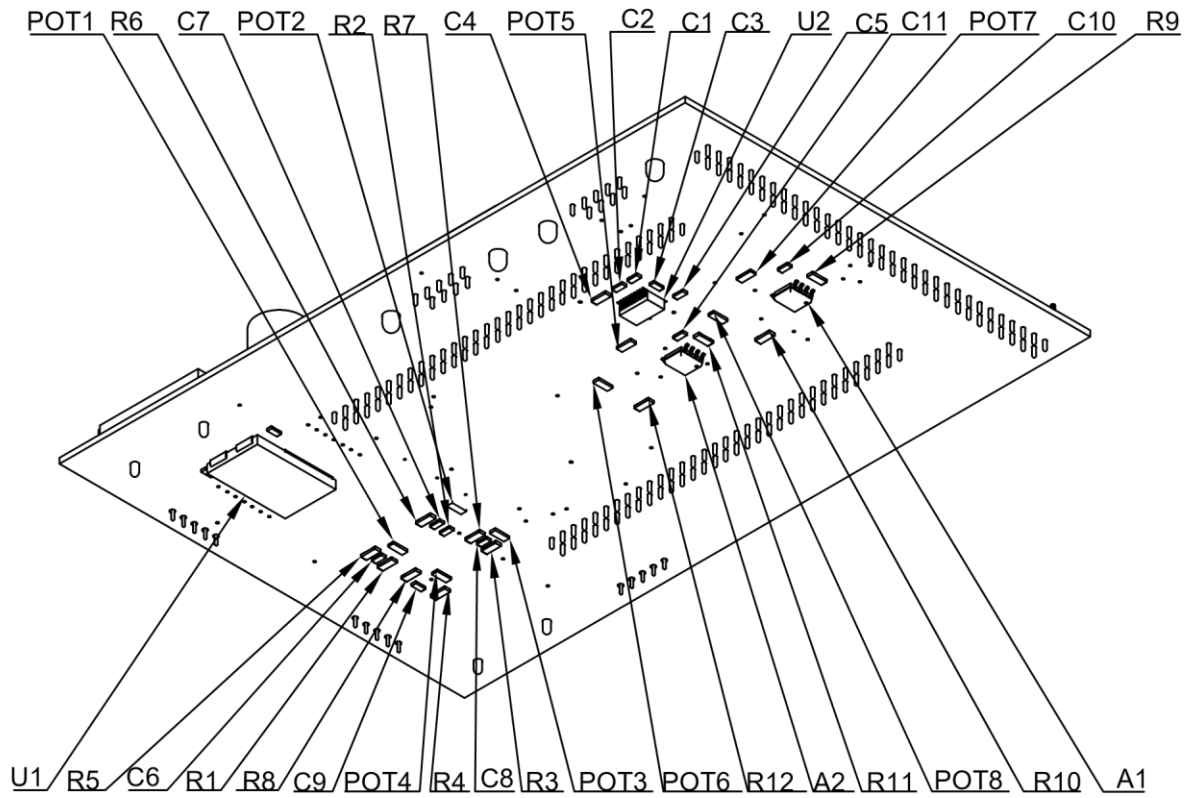
	<i>Fecha</i>	<i>Nombre</i>	<i>Firma:</i>	<i>ESCUELA TÉCNICA SUPERIOR DE INGENIERIA DEL DISEÑO Curso: 4º</i>
<i>Dibujado</i>	19/05/2023	Walid El Bachouti Khalil		
<i>Comprobado</i>				
<i>id. s. normas</i>				
<i>Escala:</i>	1:1	<i>Descripción:</i>  TOP SILK		
<i>Lamina:</i>	Nº5			



	<i>Fecha</i>	<i>Nombre</i>	<i>Firma:</i>	<i>ESCUELA TÉCNICA SUPERIOR DE INGENIERIA DEL DISEÑO Curso: 4º</i>
<i>Dibujado</i>	19/05/2023	Walid El Bachouti Khalil		
<i>Comprobado</i>				
<i>id. s. normas</i>				
<i>Escala:</i>	1:1		<i>Descripción: Bottom Silk</i>	
<i>Lamina:</i>	Nº6			



	<i>Fecha</i>	<i>Nombre</i>	<i>Firma:</i>	<i>ESCUELA TÉCNICA SUPERIOR DE INGENIERIA DEL DISEÑO Curso: 4º</i>
<i>Dibujado</i>	19/05/2023	Walid El Bachouti Khalil		
<i>Comprobado</i>				
<i>id. s. normas</i>				
<i>Escala:</i>	1:1		<i>Descripción:</i>	<i>Vista Isométrica NE</i>
<i>Lamina:</i>	Nº7			



	<i>Fecha</i>	<i>Nombre</i>	<i>Firma:</i>	<i>ESCUELA TÉCNICA SUPERIOR DE INGENIERIA DEL DISEÑO</i>  <i>Curso: 4º</i>
<i>Dibujado</i>	19/05/2023	Walid El Bachouti Khalil		
<i>Comprobado</i>				
<i>id. s. normas</i>				
<i>Escala:</i>	1:1		<i>Descripción:</i>	<i>Vista Isométrica SE</i>
<i>Lamina:</i>	Nº 8			







UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



# **PLIEGO DE CONDICIONES**

**Implementación de Máquina de  
Estados Finita en Microcontrolador  
para optimizar el control del sistema  
de energía de un dron híbrido “pila  
de combustible – batería”**



## **ÍNDICE DEL PLIEGO DE CONDICIONES**

<b>1. OBJETO</b> .....	1
<b>2. NORMATIVA GENERAL</b> .....	2
<b>3. CONDICIONES GENERALES</b> .....	3
• 3.1. Condiciones Facultativas.....	3
• 3.2 Condiciones Económicas .....	5
• 3.3 Condiciones Legales .....	6
<b>4. CONDICIONES TÉCNICAS PARTICULARES</b> .....	9
• 4.1 Placa de interfaz de hardware.....	9
4.1.1 Diseño y especificaciones .....	9
4.1.2. Requisitos del Ordenador.....	9
4.1.3. Soldadura y Componentes .....	10
4.1.4. Compatibilidad electromagnética de los componentes electrónicos .....	10
4.1.5. Entrega y Documentación.....	10
• 4.2 Software .....	11
4.2.1. Implementación de la máquina de estados finita (MEF).....	11
4.2.2. Funcionalidades del Software .....	11
4.2.3. Desarrollo y Pruebas .....	12
4.2.4. Entrega y Documentación.....	12
<b>5. EJECUCIÓN</b> .....	13
<b>6. PRUEBA DE SERVICIO</b> .....	15



## **1. OBJETO**

La presente especificación se refiere al conjunto de procesos necesarios para el control del sistema de alimentación de un dron que utiliza una combinación de pila de combustible y batería como fuente de energía. El proyecto incluye el diseño y montaje de una placa de interfaz de hardware, y la implementación software en un microcontrolador de una máquina de estados finita.

Todos los componentes serán comprados a empresas externas las cuales serán indicadas en la presente especificación. La producción de la placa hardware descrita en el presente documento será puramente a modo de prototipo. De esta manera, los materiales diseñados y fabricados exclusivamente para este proyecto serán únicos y no se llevarán a cabo procesos industriales complejos para su obtención. Asimismo, quedará excluida de esta especificación la producción de los materiales comprados.

El presente proyecto se ha realizado bajo la supervisión de dos tutores que pertenecen al Departamento de Ingeniería Electrónica de la Universidad Politécnica de Valencia y un profesor colaborador de la misma universidad.

El proyectista no se hace responsable de cualquier conflicto o mal funcionamiento del sistema que se haya generado por negligencia, dejadez o una ejecución indebida del proyecto.

## **2. NORMATIVA GENERAL**

Para realizar el pliego de condiciones se debe tener en cuenta las normativas de seguridad eléctrica, como la norma IEC 60950 o la norma IEC 62368, que establecen requisitos para la seguridad de equipos eléctricos y electrónicos de uso general.

También se deben tener en cuenta las normativas de compatibilidad electromagnética (EMC): Las normativas de EMC aseguran que los dispositivos electrónicos no interfieran entre sí y cumplan con los límites de emisiones electromagnéticas establecidos.

### **3. CONDICIONES GENERALES**

Las condiciones generales se dividen en condiciones facultativas, económicas y legales.

#### **3.1. Condiciones Facultativas**

##### **Promotor del proyecto:**

El promotor del proyecto es el tutor del Trabajo Final de Grado (TFG) perteneciente al Departamento de Ingeniería Electrónica de la Escuela Técnica Superior de Ingeniería de Diseño (ETSID) de la Universidad Politécnica de Valencia (UPV). La evaluación final del TFG se realizará por el director del proyecto y el tribunal de Evaluación de Trabajos Finales de Grado de la ETSID.

##### **Controles de calidad y ensayos:**

El promotor debe proporcionar pruebas reales que demuestren el correcto funcionamiento del sistema. Estas pruebas, acompañadas de los datos obtenidos, servirán como evidencia de que el sistema cumple con los objetivos.

##### **Recepción provisional del programa:**

El promotor recibirá una copia del programa antes de la presentación del proyecto en la ETSID.

##### **Plazo de entrega:**

El plazo de entrega es el período de dos cursos académicos otorgados para completar el TFG.

##### **Obligaciones y derechos del contratista:**

##### **Corresponde al contratista:**

- Confirmar por escrito que la documentación proporcionada es suficiente para comprender el funcionamiento del dron.
- Dirigir la ejecución material de acuerdo con el proyecto, las normas técnicas y las reglas de construcción adecuadas.



- Elaborar el Plan de Seguridad y Salud de la instalación y tomar medidas preventivas para garantizar la seguridad del personal.
- Asegurar la calidad de los materiales utilizados.
- Proporcionar al Ingeniero Técnico la lista de materiales necesarios.
- Poseer por escrito las especificaciones y documentos relevantes del proyecto.
- Acordar con el Promotor las actas de recepción provisional y definitiva.
- Conocer y cumplir las leyes relacionadas con su actividad profesional.
- Contratar seguros de accidentes de trabajo y daños a terceros durante la instalación.
- Designar un representante en la instalación con facultades para tomar decisiones.
- Acordar con el Ingeniero Técnico el acta de replanteo de la obra.
- No se aceptarán reclamaciones por falta de concordancia entre la Memoria del Proyecto y los demás documentos del proyecto.
- El sistema de monitorización debe cumplir los requisitos mencionados en la memoria.
- Verificar el buen funcionamiento del equipo antes de la entrega definitiva.
- Obligaciones y derechos del proyectista y director de obra:

**Corresponde al proyectista y director de obra:**

- Realizar el proyecto según las condiciones del contratista.
- Realizar modificaciones, adiciones o rectificaciones del proyecto según sea necesario.
- Respetar los derechos de autor.
- Consultar con el contratista cualquier modificación o solución alternativa.
- Informar periódicamente al promotor sobre el estado del proyecto.
- Cumplir con la legislación vigente.
- Cumplir con las normativas generales y específicas que afecten al proyecto.
- Disponer del material necesario para el proyecto.
- Resolver problemas técnicos imprevistos durante la ejecución del proyecto.
- Proporcionar especificaciones del programa con anticipación.
- Dirigir la obra y coordinarla con el proyecto de ejecución material.

- Asistir a las instalaciones según sea necesario.

### **3.2 Condiciones Económicas**

#### **Facultades del promotor del proyecto:**

Dado que se trata de un Trabajo Final de Grado (TFG) destinado a un proyecto de investigación, no se prevé el pago de honorarios. Sin embargo, todos los gastos se tendrán en cuenta al elaborar el presupuesto.

#### **Seguros:**

El contratista deberá contar con los seguros necesarios para cubrir posibles daños en los equipos durante la ejecución del proyecto, así como un seguro de responsabilidad civil. Además, el contratista estará obligado a asegurar la obra durante todo el período de ejecución, con una cuantía proporcional al valor de los objetos asegurados. En caso de incidentes, cualquier pago realizado por la compañía aseguradora se realizará a nombre del propietario. Los riesgos asegurados y las condiciones de la póliza se comunicarán al propietario para obtener su conformidad.

#### **Precios:**

#### **Composición de precios unitarios:**

El cálculo de los precios de las diferentes unidades se obtiene sumando los costes directos, indirectos, los gastos generales y el beneficio industrial.

#### **Se considerarán costes directos:**

- Mano de obra directamente involucrada en la ejecución de la unidad de obra.
- Materiales integrados en cada unidad o necesarios para su construcción, a precios de mercado.
- Equipos y sistemas técnicos de seguridad e higiene para la prevención y protección de accidentes y enfermedades profesionales.
- Gastos de personal, combustible, energía, etc., utilizados en el funcionamiento de los aparatos e instalaciones empleados en la ejecución de la unidad de obra.

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

- Gastos de amortización y conservación de instalaciones, sistemas y equipos mencionados anteriormente.

### **Se considerarán costes indirectos:**

- Gastos de instalación de oficinas en el lugar de trabajo, comunicaciones, construcción de almacenes, talleres, laboratorios, seguros, etc.
- Personal técnico y administrativo que participa de forma indirecta en el proyecto.

### **Beneficio industrial:**

El beneficio industrial del contratista se establece en un 6% sobre la suma total de los puntos anteriores.

### **Precio de ejecución material:**

El precio de ejecución material es la suma de los puntos anteriores, excepto el beneficio industrial.

### **Precio final:**

El precio final es la suma de los costes directos, indirectos, gastos generales y beneficio industrial.

A esta suma se le aplicará un IVA del 21%, que no está incluido en el precio.

## **3.3 Condiciones Legales**

Este capítulo aborda las disposiciones legales generales del proyecto, las responsabilidades y derechos del Contratista, el Contrato y los Impuestos.

### **Contratista:**

El Contratista debe tener la capacidad de llevar a cabo el proyecto de manera adecuada y siguiendo las indicaciones establecidas en los pliegos de condiciones, con el fin de lograr el resultado final requerido según lo especificado en la memoria.

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

Una vez finalizado el proyecto, el Contratista debe proporcionar un técnico capacitado para enseñar al personal de la empresa promotora cómo utilizar y controlar la maquinaria.

Todas las subcontratas deben cumplir los mismos criterios establecidos para el contrato principal.

### **Contrato:**

El contrato debe incluir el presupuesto total para la realización del proyecto. Además, el contrato siempre debe formalizarse por escrito y cumplir con todos los requisitos legales aplicables.

### **Responsabilidades del contratista:**

En esta sección se especifican las responsabilidades que el Contratista debe asumir durante la vigencia del contrato, incluyendo:

- Solvencia económica para pagar los materiales, la mano de obra y otros gastos.
- Garantizar una buena calidad de ejecución.
- Cumplir con las obligaciones legales y laborales relacionadas con el personal.

### **Impuestos:**

Se exigirá que el Contratista posea las licencias y certificados necesarios para llevar a cabo el proyecto.

Asimismo, deberá estar al día en el pago de impuestos, tasas y contribuciones necesarias para el desarrollo regular de las actividades de la empresa.

Todo el personal contratado debe estar legalmente registrado y dado de alta en la Seguridad Social.

El precio del producto estará sujeto al impuesto sobre el valor añadido (IVA) del 21%.

### **Causas de rescisión del proyecto:**

El promotor del proyecto puede rescindir el contrato con el proyectista en los siguientes casos:

- Retraso excesivo en la ejecución del proyecto.
- Abandono del proyecto sin causa justificada.
- Fallecimiento del proyectista.
- Causas administrativas.
- Acuerdo mutuo entre las partes, siempre y cuando ninguna de ellas se considere gravemente perjudicada.

**Condiciones legales respecto a la UPV:**

El autor declara ser el titular de los derechos de propiedad intelectual relacionados con el trabajo objeto de este acuerdo. Antes de cualquier publicación o divulgación de los documentos, se debe consultar siempre al autor.

Ambas partes se comprometen a informar a la otra parte, tan pronto como tengan conocimiento, sobre cualquier reclamación de terceros relacionada con el trabajo.

## **4. CONDICIONES TÉCNICAS PARTICULARES**

Este Pliego de Condiciones Técnicas Particulares comprende el conjunto de características que deberán cumplir los equipos empleados.

### **a. Placa de interfaz de hardware**

El diseño y montaje de una placa de interfaz de hardware para la conexión de la batería y las diferentes señales provenientes de la pila de combustible debe tener en cuenta las siguientes consideraciones:

#### **4.1.1 Diseño y especificaciones**

- a) La placa de desarrollo utilizada debe ser el STM32F429I-DISC1, de la empresa STMicroelectronics, por lo que la placa de interfaz debe disponer de 2 conectores GPIO de 64 pines en paralelo para su conexión.
- b) Se debe diseñar la placa de interfaz de hardware con dimensiones de 150x100 mm, que permita la correcta colocación de los diferentes componente y conectores.
- c) La placa de interfaz de hardware debe contar con las conexiones necesarias para la conexión de la batería y las diferentes señales provenientes de la pila de combustible.
- d) Se debe considerar la disponibilidad de puertos de comunicación UART, para la correcta transmisión de datos al ordenador.

#### **4.1.2. Requisitos del Ordenador**

- a) El ordenador utilizado para el desarrollo y programación de la placa de interfaz de hardware debe tener una pantalla mínima de 14 pulgadas, para facilitar la visualización y trabajo con los programas y herramientas de desarrollo.
- b) El ordenador debe contar con al menos dos entradas USB, para permitir la conexión de la placa de interfaz y otros dispositivos necesarios para el proyecto.

### **4.1.3. Soldadura y Componentes**

- a) Los componentes de la placa de interfaz de hardware deben ser del tipo Surface Mount Device (SMD), para permitir un diseño compacto y eficiente.
- b) La soldadura de los componentes debe realizarse utilizando estaño y una punta de soldador fina, asegurando una conexión sólida y confiable.
- c) Se recomienda adquirir los componentes necesarios para el proyecto en proveedores confiables, como Farnell o RS, para garantizar la calidad y disponibilidad de estos.

### **4.1.4. Compatibilidad electromagnética de los componentes electrónicos**

Los componentes electrónicos del sistema deberán estar regulados por el Real Decreto Ley 1580/2006 del 22 de diciembre. Esta ley regula la compatibilidad electromagnética de los equipos electrónicos que puedan crear perturbaciones electromagnéticas, o en caso contrario, que su funcionamiento pueda verse perjudicado por dichas perturbaciones.

Así, el sistema debe cumplir con un nivel legal de compatibilidad electromagnética con el fin de garantizar la no intromisión en el resto de espacio electromagnético.

### **4.1.5. Entrega y Documentación**

- a) Se debe entregar el diseño finalizado de la placa de interfaz en formato electrónico, incluyendo los archivos del esquema eléctrico, el diseño de la placa y cualquier otra documentación relevante.
- b) Se debe proporcionar una documentación completa de los componentes utilizados, así como cualquier requisito especial para la producción y montaje de la placa de interfaz.

## **4.2 Software**

Para la realización del software de control en lenguaje C se requiere formación previa del usuario:

- El usuario deberá tener conocimientos de Windows.
- El usuario deberá tener conocimientos mínimos de programación en C++, y en la programación de microcontroladores.
- Es obligación del usuario la correcta introducción de los datos y parámetros.

### **4.2.1. Implementación de la máquina de estados finita (MEF)**

- a) El software de control será implementado utilizando el entorno de desarrollo STM32CubeIDE, que es compatible con el microcontrolador STM32F429I-DISC1.
- b) Se debe implementar una máquina de estados finita (MEF) para controlar el sistema de alimentación del dron, considerando las diferentes etapas y transiciones del sistema en función de las condiciones de la pila de combustible y la batería.
- c) La MEF debe ser diseñada de manera modular y escalable, permitiendo la adición o modificación de estados y transiciones de manera sencilla para futuras mejoras o adaptaciones del sistema.

### **4.2.2. Funcionalidades del Software**

- a) El software debe ser capaz de controlar la activación y desactivación de la pila de combustible, así como monitorear su estado operativo, como nivel de carga o disponibilidad de combustible.
- b) Se debe implementar un algoritmo de gestión de la batería que monitoree su nivel de carga, evitando descargas profundas y prolongadas que puedan dañarla, y asegurando su recarga o sustitución oportuna.
- c) El software debe permitir una conmutación eficiente y suave entre la pila de combustible y la batería, priorizando el uso de la energía disponible de manera óptima y considerando los criterios establecidos para el cambio de fuente de energía.



d) Se deben implementar medidas de seguridad adecuadas para prevenir el funcionamiento en condiciones no seguras o fuera de los parámetros establecidos, como límites de temperatura o voltaje.

### **4.2.3. Desarrollo y Pruebas**

a) El desarrollo del software se realizará siguiendo las directrices proporcionadas por STM32CubeIDE para la programación del microcontrolador STM32F429I-DISC1.

b) Se realizarán pruebas exhaustivas para verificar el correcto funcionamiento del software y asegurar su compatibilidad con el hardware y la placa de interfaz diseñada.

c) Las pruebas se llevarán a cabo en diferentes escenarios y condiciones operativas, para validar el comportamiento de la MEF y garantizar su estabilidad y eficiencia en diversas situaciones.

### **4.2.4. Entrega y Documentación**

a) El software de control desarrollado, junto con cualquier biblioteca o módulo adicional utilizado, se entregará en un formato adecuado para su instalación en el microcontrolador STM32F429I-DISC1.

b) Se proporcionará una guía de usuario detallada que describa las funcionalidades del software, los procedimientos de instalación y configuración, y cualquier requisito adicional para su correcto uso y mantenimiento.

c) Se entregará una documentación técnica completa que explique el diseño y la arquitectura del software, así como los diagramas de la MEF implementada y las especificaciones técnicas relacionadas.

## **5. EJECUCIÓN**

### **1. Introducción**

En esta sección se detallan las instrucciones y pasos necesarios para la correcta ejecución del proyecto, que consiste en el diseño e implementación de una PCB y la integración de sensores de corriente y una placa de desarrollo STM32F429I-DISC1.

### **2. Conexión de los sensores de corriente**

- Identificar los conectores JST en la PCB donde se conectarán los sensores de corriente. Estos conectores deben estar debidamente marcados y su ubicación indicada en el diseño de la PCB.
- Insertar los sensores de corriente en los conectores JST correspondientes, asegurándose de que la polaridad es correcta. El sensor debe conectarse de forma adecuada, siguiendo las indicaciones de polaridad positiva y negativa.

### **3. Conexión de la batería**

- Identificar los conectores banana, donde se realizará la conexión de la batería. Los terminales deben estar claramente marcados como negativo (color negro) y positivo (color rojo) en el diseño de la PCB.
- Conectar el terminal negativo de la batería al conector banana negro y el terminal positivo al conector banana rojo. Asegurarse de que la conexión es segura y firme.

### **4. Conexión de las tensiones de celda de la batería**

- Identificar el conector JST en la PCB donde se conectarán las tensiones de celda de la batería. El conector JST debe estar marcado y su ubicación indicada en el diseño de la PCB.
- Conectar las tensiones de celda de la batería al conector JST correspondiente. Asegurarse de que la conexión se realiza correctamente, respetando la polaridad y evitando cortocircuitos.

### **5. Integración de la placa de desarrollo STM32F429I-DISC1**

- Identificar los conectores GPIO en la PCB donde se integrará la placa de desarrollo STM32F429I-DISC1. Los pines de conexión deben estar claramente marcados y su ubicación indicada en el diseño de la PCB.
- Insertar la placa de desarrollo STM32F429I-DISC1 en los conectores GPIO correspondientes, asegurándose de que el pin de alimentación coincide con el pin conectado a 5V. Verificar visualmente que todos los pines están correctamente alineados y conectados.

## **6. Programación y configuración del microcontrolador**

- Antes de realizar la conexión del microcontrolador, cargar el programa de control en la placa de desarrollo STM32F429I-DISC1. Utilizar el entorno de desarrollo STM32CubeIDE o el método de programación adecuado para cargar el firmware en el microcontrolador.
- Modificar los parámetros necesarios en el programa de control, asegurándose de ajustar los valores correctos en las variables #define. Estos parámetros incluyen las ganancias y los límites de seguridad que el usuario desea establecer.

## **7. Verificación y pruebas**

- Antes de poner en funcionamiento el sistema, realizar pruebas y verificaciones para asegurarse de que todas las conexiones y configuraciones se han realizado correctamente.
- Comprobar el correcto funcionamiento de los sensores de corriente, la placa de desarrollo y el programa de control.

## **6. PRUEBA DE SERVICIO**

### **1. Objetivo**

El objetivo de la prueba de servicio es verificar el correcto funcionamiento y rendimiento del sistema diseñado en el Trabajo de Fin de Grado (TFG). Se realizarán diferentes pruebas para comprobar que los componentes y las funcionalidades cumplen con los requisitos establecidos. Las pruebas se centrarán en los siguientes aspectos:

- Encendido de la luz incorporada en los sensores de corriente.
- Encendido del microcontrolador al conectar la batería con una tensión entre 8 y 35 V.
- Verificación de las salidas del microcontrolador mediante el uso de un multímetro y una fuente de alimentación para introducir las entradas.

### **2. Procedimiento de prueba**

A continuación, se detalla el procedimiento para llevar a cabo la prueba de servicio:

#### **2.1 Encendido de la luz incorporada en los sensores de corriente**

- Verificar que los sensores de corriente están correctamente conectados a la PCB.
- Conectar la batería al sistema, asegurándose de que la tensión está dentro del rango especificado (entre 8 y 35 V).
- Observar si la luz incorporada en los sensores de corriente se enciende. Esto indica que la conexión eléctrica y la alimentación de los sensores son correctas.

#### **2.2. Encendido del microcontrolador al conectar la batería**

- Conectar la batería al sistema, asegurándose nuevamente de que la tensión está dentro del rango especificado.
- Verificar que el microcontrolador se enciende y muestra signos de funcionamiento, como la activación de indicadores o el inicio del programa de control.

### **2.3. Verificación de las salidas del microcontrolador**

- Conectar una fuente de alimentación al sistema y establecer los valores de entrada deseados.
- Utilizar un multímetro para medir las salidas del microcontrolador.
- Verificar que las salidas se ajustan a los valores esperados de acuerdo con las entradas proporcionadas por la fuente de alimentación.

### **3. Resultados y criterios de aceptación**

- La luz incorporada en los sensores de corriente debe encenderse correctamente al conectar la batería.
- El microcontrolador debe encenderse y mostrar signos de funcionamiento cuando se le conecta la batería dentro del rango de tensión especificado.
- Las salidas del microcontrolador deben coincidir con los valores deseados establecidos por la fuente de alimentación, verificados mediante el uso del multímetro.

### **4. Conclusiones**

Los resultados obtenidos durante la prueba de servicio deben ser documentados, incluyendo observaciones relevantes y conclusiones sobre el funcionamiento del sistema. En caso de que alguna de las pruebas no cumpla con los criterios de aceptación, se deben identificar las posibles causas y tomar las medidas necesarias para corregir los problemas identificados.





UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



# **PRESUPUESTO**

**Implementación de Máquina de  
Estados Finita en Microcontrolador  
para optimizar el control del sistema  
de energía de un dron híbrido “pila  
de combustible – batería”**





## **INDICE DEL PRESUPUESTO**

<b>CAPÍTULO 1. ALCANCE .....</b>	<b>1</b>
<b>CAPÍTULO 2. COSTE MATERIAL .....</b>	<b>2</b>
<b>CAPÍTULO 3. GASTOS DE PERSONAL .....</b>	<b>5</b>
<b>CAPÍTULO 4. AMORTIZACIÓN DEL EQUIPO INFORMÁTICO .....</b>	<b>7</b>
<b>CAPÍTULO 5. PRESUPUESTO DE EJECUCIÓN MATERIAL .....</b>	<b>8</b>
<b>CAPÍTULO 6. PRESUPUESTO DE EJECUCIÓN POR CONTRATA .....</b>	<b>9</b>
<b>CAPÍTULO 7. PRESUPUESTO BASE DE LICITACIONES .....</b>	<b>10</b>



## **ÍNDICE DE TABLAS PRESUPUESTADAS**

<b>PRESUPUESTO PARCIAL N°1. Coste material .....</b>	<b>4</b>
<b>PRESUPUESTO PARCIAL N°2. Gastos del personal .....</b>	<b>6</b>
<b>PRESUPUESTO PARCIAL N°3. Amortización equipo informático.....</b>	<b>7</b>
<b>PRESUPUESTO N°1. Presupuesto Ejecución Material .....</b>	<b>8</b>
<b>PRESUPUESTO N°2. Presupuesto de Ejecución por Contrata .....</b>	<b>9</b>
<b>PRESUPUESTO N°3. Presupuesto Base de Licitaciones .....</b>	<b>10</b>



## **CAPÍTULO 1. ALCANCE**

El objetivo del presupuesto del presente Trabajo de Fin de Grado (TFG) es estimar los costes económicos del diseño de la interfaz de hardware y del software de control del sistema de energía de un dron híbrido “pila de combustible-batería”.

En primer lugar, se han incluido los costes de amortización del equipo informático que se ha empleado para la realización del proyecto, además del montaje de los sensores, la batería, y los componentes electrónicos y conectores para el diseño y montaje de la interfaz de hardware. También se han incluido los gastos de personal, basándose en los criterios del SGI – UPV (Servicio de Gestión I+D+i de la Universidad Politécnica de Valencia).

En el presente presupuesto no se consideran los costes de la carga electrónica empleada para simular el comportamiento del dron, ni los potenciómetros y placas de prueba para la simulación de la pila de combustible. Estos materiales pertenecen al laboratorio del Departamento de Ingeniería Electrónica de la UPV. Por último, tampoco se consideran los costes del gasto de personal dedicado al diseño de la placa de circuito impreso con el sensor de corriente ACS712.

## **CAPÍTULO 2. COSTE MATERIAL**

En este capítulo, se abordan los costes del material que ha sido necesario para el diseño y montaje de la placa de interfaz de hardware para el control optimizado del sistema de energía de un dron híbrido “pila de combustible - batería”.

Los precios descompuestos de la placa se dividen en los siguientes apartados:

- [1] Batería y su acondicionamiento de señal
- [2] Sensores de corriente y su acondicionamiento de señal
- [3] Comunicación RS232
- [4] Filtro Paso - Bajo
- [5] Microcontrolador
- [6] Material placa de circuito impreso

### **APARTADO 1. Batería y su acondicionamiento de señal**

<b>Nº</b>	<b>Uds</b>	<b>Descripción</b>	<b>Precio unit. (€)</b>	<b>Total (€)</b>
B1	1	Batería LiPo Gens ace 3300mAh / 14.8V	51,29	51,29
B2	1	Regulador de tensión L7805CD2T-TR	0,53	0,53
B3	1	Condensador electrolítico 100 µF	0,36	0,36
B4	1	Condensador cerámico multicapa 330 nF	0,06	0,06
B5	1	Condensador cerámico multicapa 100 nF	0,07	0,07
B6	4	Potenciómetro para PCB de 1 kΩ	0,37	1,5
B7	1	Resistencia SMD de 1 kΩ	0,01	0,01
B8	1	Resistencia SMD de 2.7 kΩ	0,02	0,02
B9	2	Resistencia SMD de 3.3 kΩ	0,03	0,06
B10	2	Conector Banana de prueba	0,77	1,54
B11	1	Carcasa de conector JST XHP-5	0,11	0,11
B12	1	Conector macho para PCB JST	0,4	0,4
B13	5	Contacto de crimpado hembra JST	0,02	0,1
			<b>TOTAL</b>	<b>56,05</b>

## APARTADO 2. Sensores de corriente y su acondicionamiento

Nº	Uds	Descripción	Precio unit. (€)	Total (€)
C1	1	Sensor de corriente ACS712 - 30A	4,73	4,73
C2	1	PCB 25x25 mm	1,00	1,00
C3	1	Cable silicona 18 AWG de 100mm	3,66	3,66
C4	1	Banana macho 3.5mm	0,07	0,07
C5	1	Banana hembra 3.5mm	0,07	0,07
C6	1	30cm cable 3 hilos apantallado	3,03	3,03
C7	3	Pin macho regleta s 2.54mm	0,01	0,03
C8	1	Condensador C1 SMD 0805 0.1 $\mu$ F	0,06	0,06
C9	1	Condensador C3 SMD 0805 1 nF	0,06	0,06
C10	1	Diodo LED D1 SMD 0805	0,16	0,16
C11	1	Resistencia R1 SMD 0603 330 $\Omega$	0,02	0,02
C12	1	Resistencia fija 470 $\Omega$	0,05	0,05
C13	1	Potenciómetro para PCB de 1 k $\Omega$	0,37	0,37
C14	1	Potenciómetro para PCB de 20 k $\Omega$	1,10	1,10
C15	1	Amplificador AD623	5,25	5,25
C16	1	Carcasa de conector JST XHP-5	0,11	0,11
C17	1	Conector macho para PCB JST	0,40	0,40
C18	3	Contacto de crimpado hembra JST	0,02	0,06
			<b>TOTAL</b>	20,23

## APARTADO 3. Comunicación RS232

Nº	Uds	Descripción	Precio unit. (€)	Total (€)
R1	1	MAX13223EEUP+	3,66	3,66
R2	2	Conector DB-9 hembra	3,40	6,80
R3	1	Cable DB-9 hembra / DB-9 hembra - 1 metro	4,34	4,34
R4	1	Convertor RS232 a USB	15,24	14,24
R5	1	Condensador cerámico multicapa de 1 $\mu$ F	0,04	0,04
R6	2	Condensador cerámico multicapa de 330 nF	0,06	0,12
R7	1	Condensador cerámico multicapa de 100 nF	0,07	0,07
R8	1	Condensador cerámico multicapa de 47 nF	0,05	0,05
R9	2	Resistencia SMD de 3.3 k $\Omega$	0,03	0,06
			<b>TOTAL</b>	29,38

Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

#### APARTADO 4. Filtro Paso - Bajo

Nº	Uds	Descripción	Precio unit. (€)	Total (€)
F1	1	Resistencia fija 470 $\Omega$	0,04	0,04
F2	1	Condensador cerámico multicapa 330 nF	0,06	0,06
			<b>TOTAL</b>	0,10

#### APARTADO 5. Microcontrolador

Nº	Uds	Descripción	Precio unit. (€)	Total (€)
M1	1	STM32F429I-DISC1 - Placa de Desarrollo	27,50	27,50
M2	3	Conector hembra de 64 vías para PCB	2,61	7,83
M3	1	Cable USB - A	4,62	4,62
			<b>TOTAL</b>	35,33

#### APARTADO 6. Material placa de circuito impreso

Nº	Uds	Descripción	Precio unit. (€)	Total (€)
CI1	1	Placa Fotorresistente, 03-5128-1, Doble Cara	3,72	3,72
			<b>TOTAL</b>	3,72

Teniendo en cuenta que se han utilizado dos sensores de corriente, uno para la batería y otro para la pila de combustible simulada, y seis filtros paso-bajo, para las entradas de los seis convertidores analógico a digital del microcontrolador, el presupuesto parcial del coste del material es el siguiente:

#### PRESUPUESTO PARCIAL N°7. Coste material

Nº	Uds	Descripción	Precio unit. (€)	Total (€)
1.1	1	Batería y su acondicionamiento de señal	56,05	4,53
1.2	2	Sensores de corriente y su acondicionamiento	20,23	40,46
1.3	1	Comunicación RS232	29,38	29,38
1.4	6	Filtro Paso - Bajo	0,10	0,60
1.5	1	Microcontrolador	35,33	35,33
1.6	1	Material placa de circuito impreso	3,72	3,72
			<b>TOTAL</b>	114,02



### **CAPÍTULO 3. GASTOS DE PERSONAL**

En el presente capítulo, se contemplan los costes relacionados con el gasto del personal que realizó el proyecto según la dedicación en horas empleada en:

[7] Diseño, selección de componentes y montaje de la placa de interfaz de hardware y de los sensores de corriente

[8] Programación del microcontrolador

[9] Sesiones de seguimiento y validación del funcionamiento del sistema

Para establecer el coste directo por hora del personal investigador según los convenios del SGI-UPV (Servicio de Gestión de la I+D+i de la Universidad Politécnica de Valencia), se considera la imputación de Walid El Bachouti como Ingeniero Electrónico Industrial y Automático recién titulado, además de la imputación de Carlos Sánchez, María Isabel Rocha y Roberto Zazo como Personal Docente Investigador (PDI).

En base a las recomendaciones de la SGI-UPV en la elaboración de presupuestos, la retribución salarial de Carlos Sánchez y María Isabel Rocha se establece en 40 €/h, siendo los dos pertenecientes a la categoría Personal Docente Investigador (PDI). Roberto Zazo es perteneciente a la categoría de Profesor Colaborador, y su retribución salarial se ha establecido en 38 €/h. Por otro lado, la retribución salarial de Walid El Bachouti se establece en 20 €/h, que representa el valor de coste promedio de un Titulado Superior.

#### **APARTADO 7. Diseño, selección de componentes y montaje de la placa de interfaz de hardware y sensores de corriente**

<b>Nº</b>	<b>Horas</b>	<b>Concepto</b>	<b>Precio por hora (€)</b>	<b>Total (€)</b>
MO1	70	Titulado Superior	20,00	1400,00
MO2	15	Profesor Docente Investigador (1)	40,00	600,00
MO3	6	Profesor Docente Investigador (2)	40,00	240,00
MO4	2	Profesor colaborador	38,00	76,00
			<b>TOTAL</b>	<b>2316,00</b>

Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

#### APARTADO 8. Programación del microcontrolador

Nº	Horas	Concepto	Precio por hora (€)	Total (€)
MO1	50	Titulado Superior	20,00	1000,00
MO2	2	Profesor Docente Investigador (1)	40,00	80,00
MO3	8	Profesor Docente Investigador (2)	40,00	320,00
MO4	8	Profesor Colaborador	38,00	304,00
			<b>TOTAL</b>	1704,00

#### APARTADO 9. Sesiones de seguimiento y validación del funcionamiento del sistema

Nº	Horas	Concepto	Precio por hora (€)	Total (€)
MO1	30	Titulado Superior	20,00	600,00
MO2	15	Profesor Docente Investigador (1)	40,00	600,00
MO3	8	Profesor Docente Investigador (2)	40,00	320,00
MO4	10	Profesor Colaborador	38,00	380,00
			<b>TOTAL</b>	1900,00

Por lo tanto, el presupuesto parcial del coste de los gastos del personal es el siguiente:

#### PRESUPUESTO PARCIAL N°8. Gastos del personal

Nº	Horas	Concepto	Total (€)
2.1	150	Titulado Superior	3000,00
2.2	32	Titular de Universidad (1)	1280,00
2.3	20	Titular de Universidad (2)	880,00
2.4	18	Profesor Colaborador	760,00
			<b>TOTAL</b>
			5920,00
			<b>MANO DE OBRA INDIRECTA</b>
			5%
			<b>COSTE TOTAL MANO DE OBRA</b>
			6216,00

## **CAPÍTULO 4. AMORTIZACIÓN DEL EQUIPO INFORMÁTICO**

En el presente capítulo, por un lado, se consideran los gastos de amortización del equipo informático empleado para la realización del TFG.

Siguiendo las recomendaciones del SGI-UPV, la amortización del equipo informático se determina mediante la siguiente expresión (Ec.1):

$$AMORTIZACIÓN (\text{€}) = \frac{USO (\text{años})}{PERIODO DE AMORTIZACIÓN (\text{años})} \cdot PRECIO (\text{€})$$

Siendo el periodo de amortización de 6 años para un equipo informático. Ha sido empleado el modelo Omen 15 de la empresa HP, con un precio en el mercado de 1200 €, y el tiempo de uso del portátil para la realización de este TFG ha resultado equivalente que la duración de este, ya que se empleó tanto en la redacción de la memoria como en la programación del microcontrolador o el diseño de la tarjeta de interfaz de hardware (~ 5 meses). Por lo tanto:

### **PRESUPUESTO PARCIAL N°9. Amortización equipo informático**

<b>N°</b>	<b>Uso (meses)</b>	<b>Concepto</b>	<b>Precio (€)</b>	<b>Amortización (€)</b>
3.1	5	HP Omen 15	1.200,00	100,00
		<b>TOTAL</b>		100,00

Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

## **CAPÍTULO 5. PRESUPUESTO DE EJECUCIÓN MATERIAL**

El Presupuesto de la Ejecución del Material (PEM) es el resultado de la suma de los presupuestos parciales anteriores:

### **PRESUPUESTO N°45. Presupuesto Ejecución Material**

<b>PEM</b>	<b>Precio (€)</b>
PRESUPUESTO PARCIAL N°1. Coste material	114,02
PRESUPUESTO PARCIAL N°3. Gastos del personal	6216,00
PRESUPUESTO PARCIAL N°3. Amortización equipo informático	100,00
<b>COSTE TOTAL EJECUCIÓN MATERIAL</b>	<b>6430,02</b>

El Presupuesto de la Ejecución del Material (PEM) asciende a SEIS MIL CUATROCIENTOS TREINTA EUROS CON DOS CÉNTIMOS.

## **CAPÍTULO 6. PRESUPUESTO DE EJECUCIÓN POR CONTRATA**

En el Presupuesto de Ejecución por Contrata (PEC) se contabilizan los gastos generales como costes indirectos, los cuales incluyen los equipos de instrumentación electrónica, como multímetros, fuentes de alimentación o soldadores. También incluyen agua, luz, internet, el uso del laboratorio, el personal que mantiene dicho laboratorio, etc. Estos costes suponen un 10% del Presupuesto de Ejecución Material.

Además, también se incluye un Beneficio Industrial (BI) sobre el Presupuesto de Ejecución Material, el cual va a ser de un 6%.

### **PRESUPUESTO N°46. Presupuesto de Ejecución por Contrata**

<b>PEC</b>	<b>Total (€)</b>
Presupuesto Ejecución Material	6430,02
Gastos generales 10%	745,88
Beneficio Industrial 6%	385,82
<b>COSTE TOTAL EJECUCIÓN POR CONTRATA</b>	<b>7458,82</b>

El Presupuesto de Ejecución por Contrata (PEC) asciende a SIETE MIL CUATROCIENTOS CINCUENTA Y OCHO EUROS CON OCHENTA Y DOS CÉNTIMOS.

## **CAPÍTULO 7. PRESUPUESTO BASE DE LICITACIONES**

En el Presupuesto Base de Licitaciones (PBL) se incluye el Impuesto sobre el Valor Añadido (IVA) del 21% al Presupuesto de Ejecución por Contrata:

### **PRESUPUESTO N°47. Presupuesto Base de Licitaciones**

<b>PBL</b>	<b>Precio (€)</b>
Presupuesto de Ejecución por Contrata	7458,82
IVA 21 %	1566,35
<b>COSTE TOTAL BASE DE LICITACIONES</b>	<b>9025,17</b>

El presupuesto Base de Licitaciones (PBL) asciende a NUEVE MIL VEINTICINCO EUROS CON DIECISIETE CÉNTIMOS.





UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



# **ANEXO I**

**Máquina de estados, diagramas de flujo y código fuente**

**Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”**





## **ÍNDICE DEL ANEXO I**

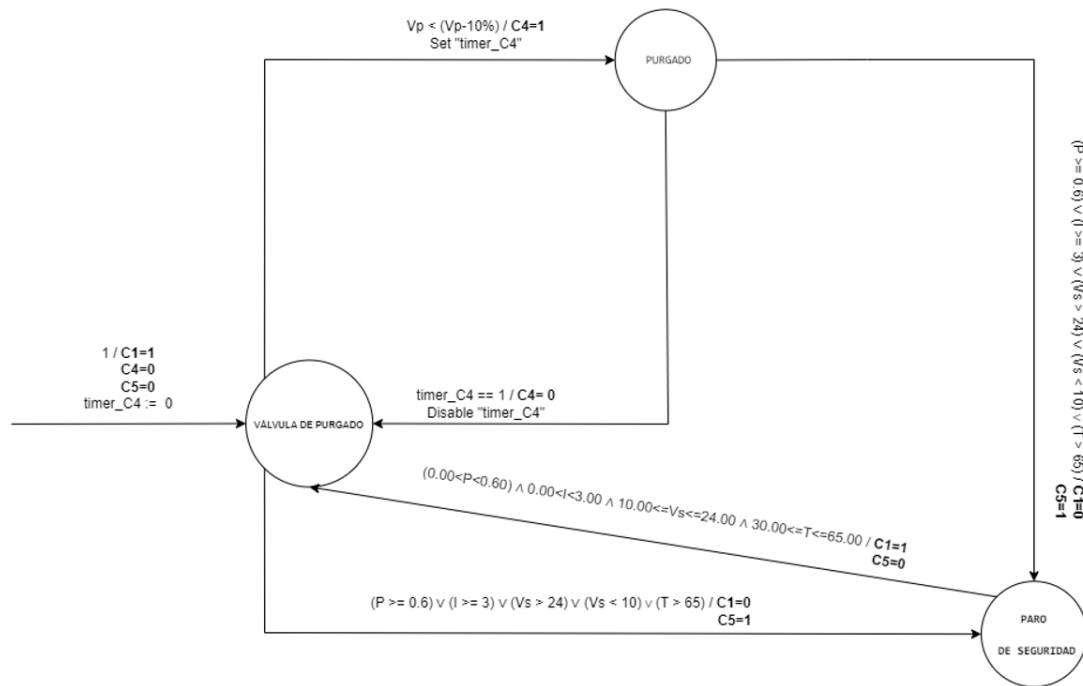
<b>1. LEYENDA GENERAL DE LA MÁQUINA DE ESTADOS .....</b>	<b>1</b>
<b>2. MÁQUINA DE ESTADOS VÁLVULA DE PURGADO .....</b>	<b>2</b>
<b>3. MÁQUINA DE ESTADOS VENTILADOR.....</b>	<b>3</b>
<b>4. MÁQUINA DE ESTADOS CONVERTIDOR DE SALIDA .....</b>	<b>4</b>
<b>5. FLUJOGRAMAS .....</b>	<b>5</b>
<b>6. CÓDIGO FUENTE .....</b>	<b>7</b>
• <b>6.1 main.c .....</b>	<b>7</b>
• <b>6.2 main.h.....</b>	<b>29</b>



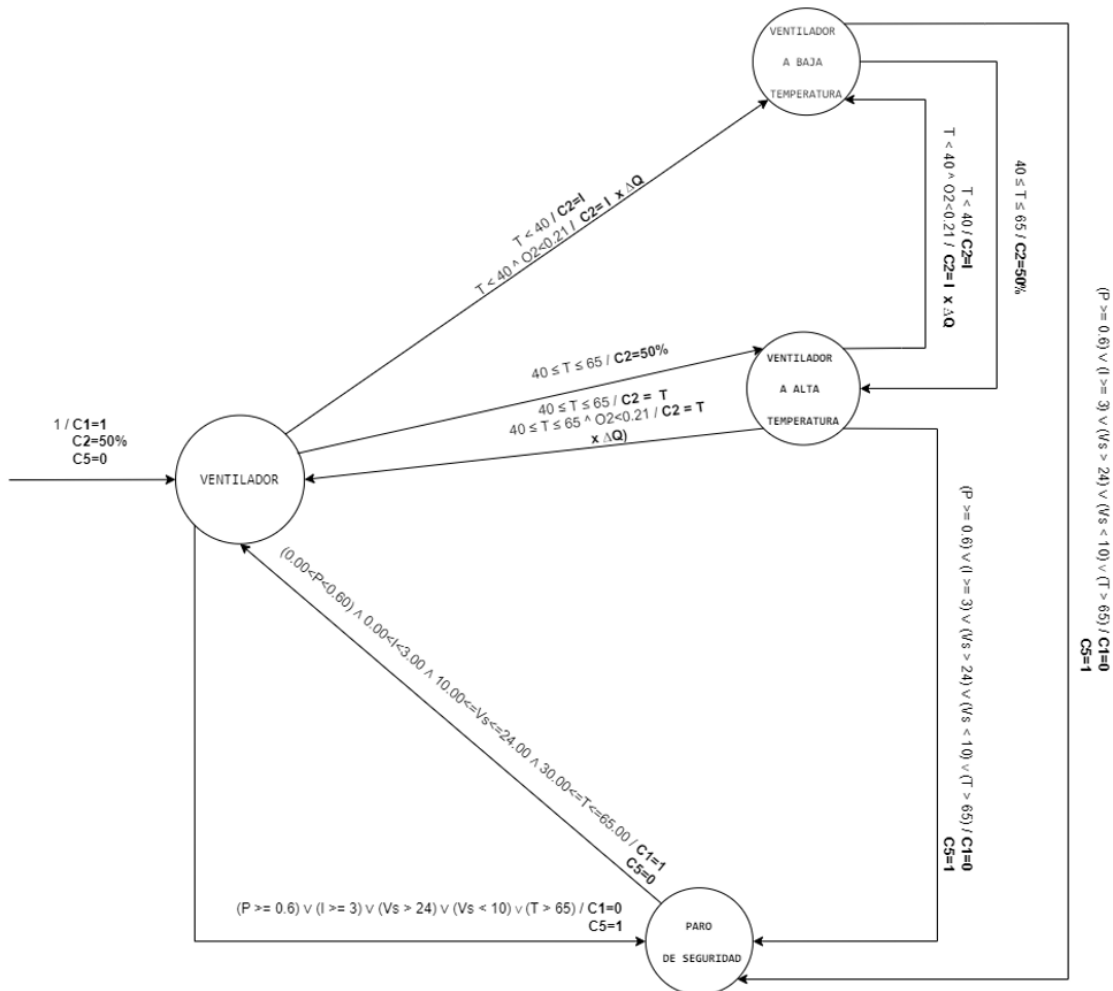
## 1. LEYENDA GENERAL DE LA MÁQUINA DE ESTADOS

• Inputs -->	- V1: R {0.00, ..., 4.13} U Absent
	- V2: R {0.00, ..., 8.29} U Absent
	- V3: R {0.00, ..., 12.44} U Absent
	- V4: R {0.00, ..., 16.59} U Absent
	- VBAT: R {0.00, ..., 16.59} U Absent
	- Im: R {0.00, ..., 4.00} U Absent
	- I: R {0.00, ..., 4.00} U Absent
	- Vp: R {0.25, ..., 1.25} U Absent
	- Vs: R {5.00, ..., 25.00} U Absent
	- T: R {30.00, ..., 70.00} U Absent
	- O2: R {0.19, ..., 0.25} U Absent
	- P: R {0.00, ..., 1.00} U Absent
	- Q: R {2.06, ..., 2.70} U Absent
• Outputs -->	- C1: N {0,1} U Absent
	- C2: R {0.00, ..., 5.00} U Absent
	- C3: R {0.00, ..., 5.00} U Absent
	- C4: N {0,1} U Absent
	- C5: N {0,1} U Absent
• Variable -->	- timer_C4 : N {0,1} U Absent

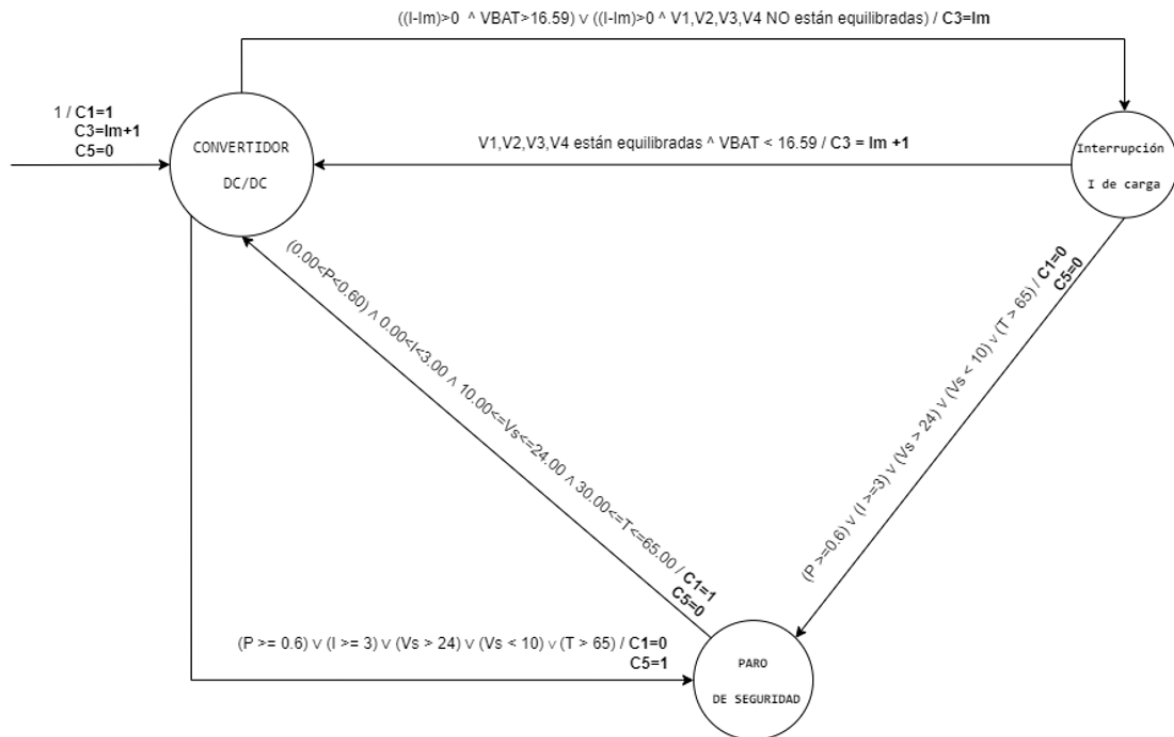
## 2. MÁQUINA DE ESTADOS VÁLVULA DE PURGADO



### 3. MÁQUINA DE ESTADOS VENTILADOR

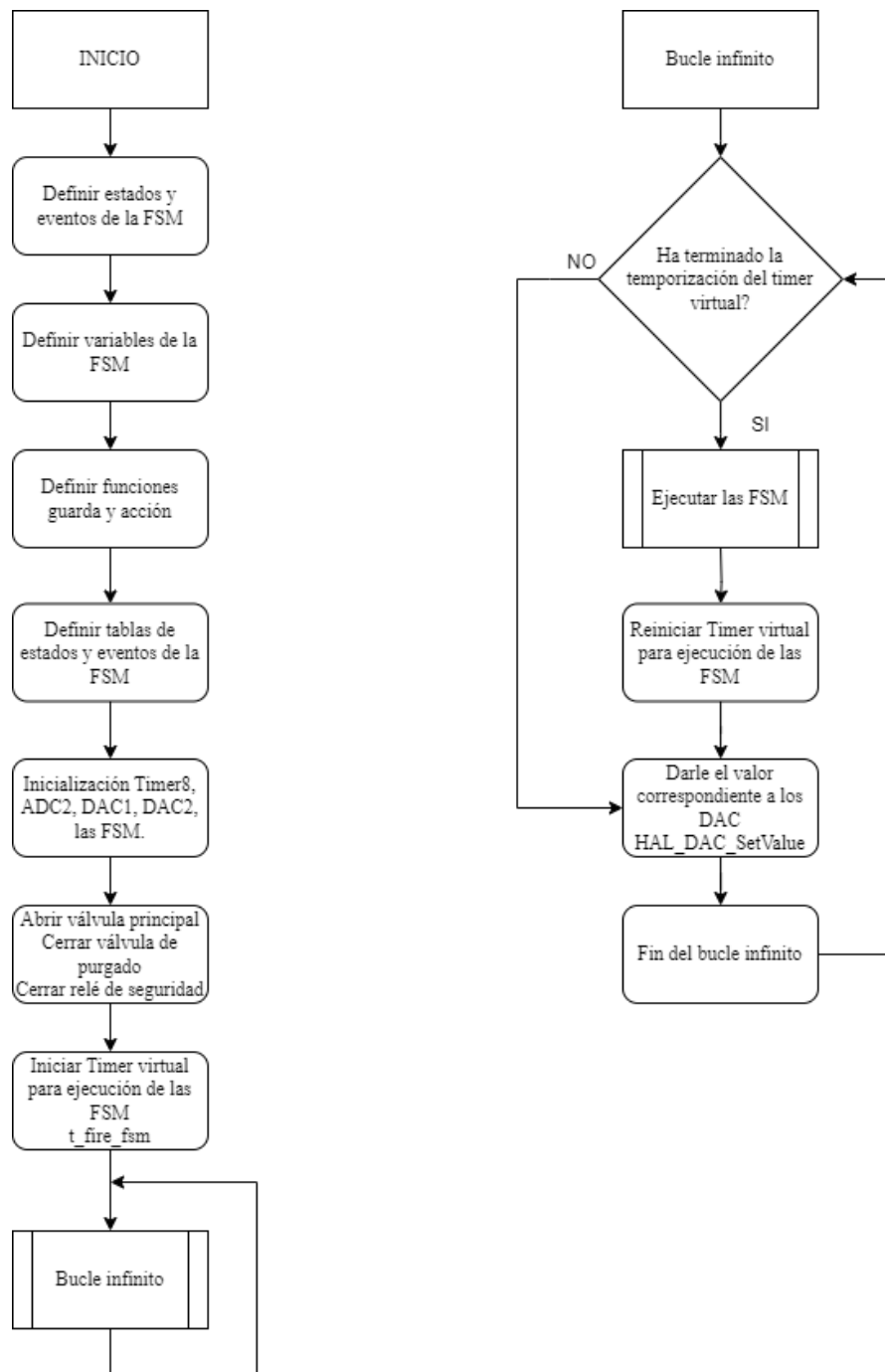


#### 4. MÁQUINA DE ESTADOS CONVERTIDOR DE SALIDA



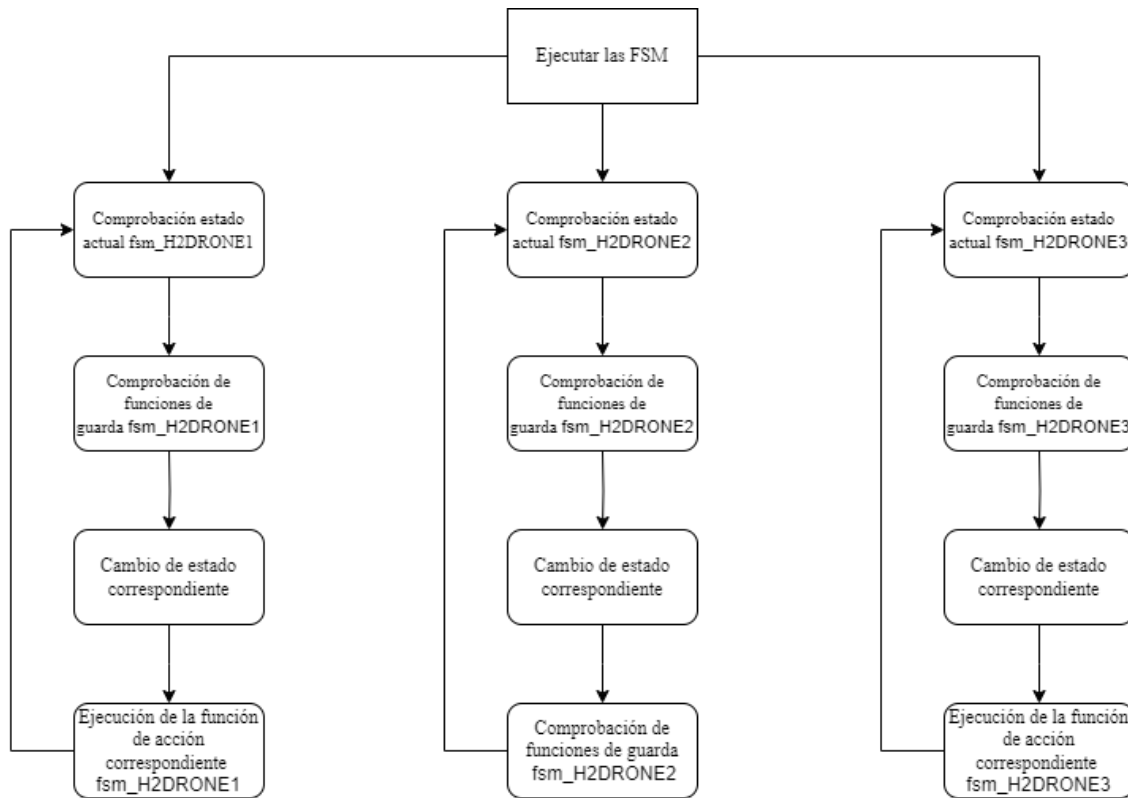
## 5. FLUJOGRAMAS

### MAIN.C





## EJECUCIÓN DE LAS MÁQUINAS DE ESTADO



## 6. CÓDIGO FUENTE

### 6.1 main.c

```
/* USER CODE BEGIN Header */
/**
*****
 * @file      : main.c
 * @brief     : Main program body
*****
 * @attention
 *
 * Copyright (c) 2023 STMicroelectronics.
 * All rights reserved.
 *
 * This software is licensed under terms that can be found in the LICENSE
file
 * in the root directory of this software component.
 * If no LICENSE file comes with this software, it is provided AS-IS.
 *
*****
 */
/* USER CODE END Header */
/* Includes ----- */
#include "main.h"

/* Private includes ----- */
/* USER CODE BEGIN Includes */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "fsm.h"
#include "vtimers.h"

/* USER CODE END Includes */

/* Private typedef ----- */
/* USER CODE BEGIN PTD */
/* USER CODE END PTD */
/* Private define ----- */
/* USER CODE BEGIN PD */
/* USER CODE END PD */
/* Private macro ----- */
/* USER CODE BEGIN PM */
/* USER CODE END PM */
/* Private variables ----- */
ADC_HandleTypeDef hadc2;
DMA_HandleTypeDef hdma_adc2;

DAC_HandleTypeDef hdac;
DMA_HandleTypeDef hdma_dac1;
```

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

```
DMA_HandleTypeDef hdma_dac2;
TIM_HandleTypeDef htim8;

UART_HandleTypeDef huart5;
UART_HandleTypeDef huart7;
DMA_HandleTypeDef hdma_uart5_rx;
DMA_HandleTypeDef hdma_uart5_tx;
DMA_HandleTypeDef hdma_uart7_rx;
DMA_HandleTypeDef hdma_uart7_tx;

/* USER CODE BEGIN PV */
/* FSM PV State Enumeration */
// State Enumeration for "H2DRONE" FSM
enum state {
    CONVERTIDOR_DCDC,
    VENTILADOR,
    VALVULA_PURGADO,
    INTERRUPCION_CARGA,
    VENTILADOR_A_ALTA_TEMPERATURA,
    VENTILADOR_A_BAJA_TEMPERATURA,
    PURGADO,
    PARO_DE_SEGURIDAD,
};

/* Declaration of inputs and variables of the FSMs */
//Timers virtuales
static struct timer timer_C4;
static struct timer t_fire_fsm;

//Inputs
static float volts[CANALES_ADC];
static float corriente[SENSORES_DE_CORRIENTE];
static float presion;
static float concentracion_O2;
static float temperatura;

//Conversiones ADC
//Vector para guardar los valores leídos
static uint16_t valor_adc[CANALES_ADC];
//Longitud
const int contador_canales_adc = sizeof(valor_adc)/sizeof(valor_adc[0]);

//DAC
uint32_t dac_value1=0;
uint32_t dac_value2=0;

//Variables globales para visualización en modo Debug
uint8_t estado_C1,estado_C4,estado_C5;
float V1,V2,V3,V4,VBAT,Vs,Vp;

/* USER CODE END PV */
/* Private function prototypes -----*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_DMA_Init(void);
```

Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

```
static void MX_UART5_Init(void);
static void MX_DAC_Init(void);
static void MX_ADC2_Init(void);
static void MX_TIM8_Init(void);
static void MX_UART7_Init(void);

/* USER CODE BEGIN PFP */
/* ----- General Purpose Functions ----- */

/* ----- Guard Functions ----- */
// H2DRONE -> Funciones de las condiciones (guardas)
static uint8_t bat_desequilibrada_or_max (fsm_t* this);
static uint8_t bat_equilibrada (fsm_t* this);
static uint8_t temperatura_alta (fsm_t* this);
static uint8_t temperatura_baja (fsm_t* this);
static uint8_t tension_de_celda_baja (fsm_t* this);
static uint8_t tiempo_apertura_C4 (fsm_t* this);
static uint8_t parametros_de_seguridad (fsm_t* this);
static uint8_t parametros_rango_normal (fsm_t* this);

/* ----- Action Functions ----- */
// H2DRONE -> Funciones de las acciones
static void corriente_de_consumo (fsm_t* this);
static void carga_normal (fsm_t* this);
static void ventilador_carga_media (fsm_t* this);
static void ventilador_proporcional_I_Q (fsm_t* this);
static void ventilador_prporcional_T_Q (fsm_t* this);
static void abrir_C4 (fsm_t* this);
static void cerrar_C4(fsm_t* this);
static void abrir_circuito(fsm_t* this);
static void cerrar_circuito(fsm_t* this);

/* USER CODE END PFP */
/* Private user code -----*/
/* USER CODE BEGIN 0 */

/* ----- EXPLICIT FSM DESCRIPTION -----*/
// TABLA ESTADOS Y TRANSICIONES H2DRONE
//FSM PARA VALVULA DE PURGADO DE LA PILA
static fsm_trans_t table_H2DRONE1[] = {
{ VALVULA_PURGADO, tension_de_celda_baja, PURGADO, abrir_C4 },
{ PURGADO, tiempo_apertura_C4, VALVULA_PURGADO, cerrar_C4 },
{ VALVULA_PURGADO, parametros_de_seguridad, PARO_DE_SEGURIDAD,
abrir_circuito},
{ PARO_DE_SEGURIDAD, parametros_rango_normal, VALVULA_PURGADO,
cerrar_circuito },
{ PURGADO, parametros_de_seguridad, PARO_DE_SEGURIDAD, abrir_circuito },
{ PARO_DE_SEGURIDAD, parametros_rango_normal, VALVULA_PURGADO,
cerrar_circuito },
{ -1, NULL, -1, NULL },
};

//FSM PARA VENTILADOR DE LA PILA
static fsm_trans_t table_H2DRONE2[] = {
```

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

```
{ VENTILADOR, temperatura_alta , VENTILADOR_A_ALTA_TEMPERATURA ,
ventilador_carga_media },
{ VENTILADOR_A_ALTA_TEMPERATURA, temperatura_baja
,VENTILADOR_A_BAJA_TEMPERATURA , ventilador_proporcional_I_Q },
{ VENTILADOR_A_BAJA_TEMPERATURA, temperatura_alta ,
VENTILADOR_A_ALTA_TEMPERATURA,ventilador_carga_media },
{ VENTILADOR_A_ALTA_TEMPERATURA, temperatura_alta, VENTILADOR,
ventilador_prporcional_I_Q },
{ VENTILADOR, parametros_de_seguridad, PARO_DE_SEGURIDAD, abrir_circuito },
{ PARO_DE_SEGURIDAD, parametros_rango_normal, VENTILADOR, cerrar_circuito },
{ VENTILADOR_A_ALTA_TEMPERATURA, parametros_de_seguridad, PARO_DE_SEGURIDAD,
abrir_circuito },
{ PARO_DE_SEGURIDAD, parametros_rango_normal, VENTILADOR, cerrar_circuito },
{ VENTILADOR_A_BAJA_TEMPERATURA, parametros_de_seguridad, PARO_DE_SEGURIDAD,
abrir_circuito },
{ PARO_DE_SEGURIDAD, parametros_rango_normal, VENTILADOR, cerrar_circuito },
{ -1, NULL, -1, NULL },
};

//FSM PARA CONVERTIDOR DC/DC DE LA SALIDA DE LA PILA
static fsm_trans_t table_H2DRONE3[] = {
{ CONVERTIDOR_DCDC, bat_desequilibrada_or_max, INTERRUPCION_CARGA,
corriente_de_consumo },
{ INTERRUPCION_CARGA, bat_equilibrada, CONVERTIDOR_DCDC, carga_normal },
{ CONVERTIDOR_DCDC, parametros_de_seguridad, PARO_DE_SEGURIDAD,
abrir_circuito },
{ PARO_DE_SEGURIDAD, parametros_rango_normal, CONVERTIDOR_DCDC,
cerrar_circuito },
{ INTERRUPCION_CARGA, parametros_de_seguridad, PARO_DE_SEGURIDAD,
abrir_circuito },
{ PARO_DE_SEGURIDAD, parametros_rango_normal, CONVERTIDOR_DCDC,
cerrar_circuito },
{ -1, NULL, -1, NULL },
};

/* USER CODE END 0 */

/**
 * @brief The application entry point.
 * @retval int
 */
int main(void)
{
    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration-----*/

    /* Reset of all peripherals, Initializes the Flash interface and the
    SysTick. */
    HAL_Init();

    /* USER CODE BEGIN Init */
    /* USER CODE END Init */
```

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

```
/* Configure the system clock */
SystemClock_Config();

/* USER CODE BEGIN SysInit */
/* USER CODE END SysInit */

/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_DMA_Init();
MX_UART5_Init();
MX_DAC_Init();
MX_ADC2_Init();
MX_TIM8_Init();
MX_UART7_Init();

/* USER CODE BEGIN 2 */

    //Inicialización Timer 8 con interrupción
    HAL_TIM_Base_Start_IT(&htim8);

    //Inicialización DAC1 y DAC2 con DMA
    HAL_DAC_Start_DMA(&hdac, DAC_CHANNEL_1, &dac_value1, 1,
    DAC_ALIGN_12B_R);
    HAL_DAC_Start_DMA(&hdac, DAC_CHANNEL_2, &dac_value2, 1,
    DAC_ALIGN_12B_R);

    //Inicialización ADC con DMA
    HAL_ADC_Start_DMA(&hadc2, (uint32_t*)valor_adc, contador_canales_adc);

/* Declaración FSMs */
fsm_t* fsm_H2DRONE1 = fsm_new(table_H2DRONE1);
fsm_t* fsm_H2DRONE2 = fsm_new(table_H2DRONE2);
fsm_t* fsm_H2DRONE3 = fsm_new(table_H2DRONE3);

/* Inicialización FSMs */
fsm_init(fsm_H2DRONE1, table_H2DRONE1);
fsm_init(fsm_H2DRONE2, table_H2DRONE2);
fsm_init(fsm_H2DRONE3, table_H2DRONE3);

//Inicialización estado válvulas y relé
HAL_GPIO_WritePin(GPIOF, GPIO_PIN_10, 0); //Cerrar válvula de purgado(C4)
HAL_GPIO_WritePin(GPIOF, GPIO_PIN_8, 1); //Abrir válvula principal(C1)
HAL_GPIO_WritePin(GPIOF, GPIO_PIN_9, 0); //Cerrar relé de seguridad(C5)

//Inicialización timer virtual para ejecutar FSM, T_FIRE es la duracion
del timer (5 ms)
timer_set(&t_fire_fsm,T_FIRE);

/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
```

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

```
//Ejecución de la FSM
    if (timer_ended(&t_fire_fsm) == 1){

        //Visualización en modo Debug
        estado_C1=HAL_GPIO_ReadPin(GPIOF, GPIO_PIN_8);
        estado_C4=HAL_GPIO_ReadPin(GPIOF, GPIO_PIN_10);
        estado_C5=HAL_GPIO_ReadPin(GPIOF, GPIO_PIN_9);
        V1=volts[7];
        V2=volts[8];
        V3=volts[9];
        V4=volts[0];
        VBAT=volts[0];
        Vs=volts[4];
        Vp=volts[6];

        // Fire FSM
        fsm_fire(fsm_H2DRONE1);
        fsm_fire(fsm_H2DRONE2);
        fsm_fire(fsm_H2DRONE3);

        //Reiniciar Timer virtual para ejecutar FSMs,
        T_FIRE es la duracion del timer (20 ms)
        timer_set(&t_fire_fsm,T_FIRE);

    }

    // Establecer valores específicos al ejecutarse las FSMs
    HAL_DAC_SetValue(&hdac, DAC_CHANNEL_1, DAC_ALIGN_12B_R, dac_value1);
    HAL_DAC_SetValue(&hdac, DAC_CHANNEL_2, DAC_ALIGN_12B_R, dac_value2);

    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */
}

/**
 * @brief System Clock Configuration
 * @retval None
 */
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

    /** Configure the main internal regulator output voltage
    */
    __HAL_RCC_PWR_CLK_ENABLE();
    __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);

    /** Initializes the RCC Oscillators according to the specified parameters
    * in the RCC_OscInitTypeDef structure.
    */
    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
```

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

```
RCC_OscInitStruct.HSEState = RCC_HSE_ON;
RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
RCC_OscInitStruct.PLL.PLLM = 4;
RCC_OscInitStruct.PLL.PLLN = 180;
RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
RCC_OscInitStruct.PLL.PLLQ = 3;
if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
{
    Error_Handler();
}

/** Activate the Over-Drive mode
*/
if (HAL_PWREx_EnableOverDrive() != HAL_OK)
{
    Error_Handler();
}

/** Initializes the CPU, AHB and APB buses clocks
*/
RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYCLK
                             |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV4;
RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV2;

if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_5) != HAL_OK)
{
    Error_Handler();
}
}

/**
 * @brief ADC2 Initialization Function
 * @param None
 * @retval None
 */
static void MX_ADC2_Init(void)
{
    /* USER CODE BEGIN ADC2_Init 0 */

    /* USER CODE END ADC2_Init 0 */

    ADC_ChannelConfTypeDef sConfig = {0};

    /* USER CODE BEGIN ADC2_Init 1 */

    /* USER CODE END ADC2_Init 1 */

    /** Configure the global features of the ADC (Clock, Resolution, Data
    Alignment and number of conversion)
    */
}
```



## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

```
hadc2.Instance = ADC2;
hadc2.Init.ClockPrescaler = ADC_CLOCK_SYNC_PCLK_DIV4;
hadc2.Init.Resolution = ADC_RESOLUTION_12B;
hadc2.Init.ScanConvMode = ENABLE;
hadc2.Init.ContinuousConvMode = DISABLE;
hadc2.Init.DiscontinuousConvMode = DISABLE;
hadc2.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_RISING;
hadc2.Init.ExternalTrigConv = ADC_EXTERNALTRIGCONV_T8_TRGO;
hadc2.Init.DataAlign = ADC_DATAALIGN_RIGHT;
hadc2.Init.NbrOfConversion = 11;
hadc2.Init.DMAContinuousRequests = ENABLE;
hadc2.Init.EOCSelection = ADC_EOC_SINGLE_CONV;
if (HAL_ADC_Init(&hadc2) != HAL_OK)
{
    Error_Handler();
}

/** Configure for the selected ADC regular channel its corresponding rank
in the sequencer and its sample time.
*/
sConfig.Channel = ADC_CHANNEL_15;
sConfig.Rank = 1;
sConfig.SamplingTime = ADC_SAMPLETIME_15CYCLES;
if (HAL_ADC_ConfigChannel(&hadc2, &sConfig) != HAL_OK)
{
    Error_Handler();
}

/** Configure for the selected ADC regular channel its corresponding rank
in the sequencer and its sample time.
*/
sConfig.Channel = ADC_CHANNEL_2;
sConfig.Rank = 2;
if (HAL_ADC_ConfigChannel(&hadc2, &sConfig) != HAL_OK)
{
    Error_Handler();
}

/** Configure for the selected ADC regular channel its corresponding rank
in the sequencer and its sample time.
*/
sConfig.Channel = ADC_CHANNEL_3;
sConfig.Rank = 3;
if (HAL_ADC_ConfigChannel(&hadc2, &sConfig) != HAL_OK)
{
    Error_Handler();
}

/** Configure for the selected ADC regular channel its corresponding rank
in the sequencer and its sample time.
*/
sConfig.Channel = ADC_CHANNEL_6;
sConfig.Rank = 4;
if (HAL_ADC_ConfigChannel(&hadc2, &sConfig) != HAL_OK)
{
```

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

```
    Error_Handler();
}

/** Configure for the selected ADC regular channel its corresponding rank
in the sequencer and its sample time.
*/
sConfig.Channel = ADC_CHANNEL_7;
sConfig.Rank = 5;
if (HAL_ADC_ConfigChannel(&hadc2, &sConfig) != HAL_OK)
{
    Error_Handler();
}

/** Configure for the selected ADC regular channel its corresponding rank
in the sequencer and its sample time.
*/
sConfig.Channel = ADC_CHANNEL_8;
sConfig.Rank = 6;
if (HAL_ADC_ConfigChannel(&hadc2, &sConfig) != HAL_OK)
{
    Error_Handler();
}

/** Configure for the selected ADC regular channel its corresponding rank
in the sequencer and its sample time.
*/
sConfig.Channel = ADC_CHANNEL_9;
sConfig.Rank = 7;
if (HAL_ADC_ConfigChannel(&hadc2, &sConfig) != HAL_OK)
{
    Error_Handler();
}

/** Configure for the selected ADC regular channel its corresponding rank
in the sequencer and its sample time.
*/
sConfig.Channel = ADC_CHANNEL_11;
sConfig.Rank = 8;
if (HAL_ADC_ConfigChannel(&hadc2, &sConfig) != HAL_OK)
{
    Error_Handler();
}

/** Configure for the selected ADC regular channel its corresponding rank
in the sequencer and its sample time.
*/
sConfig.Channel = ADC_CHANNEL_12;
sConfig.Rank = 9;
if (HAL_ADC_ConfigChannel(&hadc2, &sConfig) != HAL_OK)
{
    Error_Handler();
}

/** Configure for the selected ADC regular channel its corresponding rank
in the sequencer and its sample time.
```

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

```
*/
sConfig.Channel = ADC_CHANNEL_13;
sConfig.Rank = 10;
if (HAL_ADC_ConfigChannel(&hadc2, &sConfig) != HAL_OK)
{
    Error_Handler();
}

/** Configure for the selected ADC regular channel its corresponding rank
in the sequencer and its sample time.
*/
sConfig.Channel = ADC_CHANNEL_14;
sConfig.Rank = 11;
if (HAL_ADC_ConfigChannel(&hadc2, &sConfig) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN ADC2_Init 2 */

/* USER CODE END ADC2_Init 2 */

}

/**
 * @brief DAC Initialization Function
 * @param None
 * @retval None
 */
static void MX_DAC_Init(void)
{
    /* USER CODE BEGIN DAC_Init 0 */

    /* USER CODE END DAC_Init 0 */

    DAC_ChannelConfTypeDef sConfig = {0};

    /* USER CODE BEGIN DAC_Init 1 */

    /* USER CODE END DAC_Init 1 */

    /** DAC Initialization
    */
    hdac.Instance = DAC;
    if (HAL_DAC_Init(&hdac) != HAL_OK)
    {
        Error_Handler();
    }

    /** DAC channel OUT1 config
    */
    sConfig.DAC_Trigger = DAC_TRIGGER_NONE;
    sConfig.DAC_OutputBuffer = DAC_OUTPUTBUFFER_ENABLE;
    if (HAL_DAC_ConfigChannel(&hdac, &sConfig, DAC_CHANNEL_1) != HAL_OK)
    {
```

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

```
    Error_Handler();
}

/** DAC channel OUT2 config
*/
if (HAL_DAC_ConfigChannel(&hdac, &sConfig, DAC_CHANNEL_2) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN DAC_Init 2 */

/* USER CODE END DAC_Init 2 */

}

/**
 * @brief TIM8 Initialization Function
 * @param None
 * @retval None
 */
static void MX_TIM8_Init(void)
{
    /* USER CODE BEGIN TIM8_Init 0 */

    /* USER CODE END TIM8_Init 0 */

    TIM_ClockConfigTypeDef sClockSourceConfig = {0};
    TIM_MasterConfigTypeDef sMasterConfig = {0};

    /* USER CODE BEGIN TIM8_Init 1 */

    /* USER CODE END TIM8_Init 1 */
    htim8.Instance = TIM8;
    htim8.Init.Prescaler = 2700;
    htim8.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim8.Init.Period = 60000;
    htim8.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim8.Init.RepetitionCounter = 0;
    htim8.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
    if (HAL_TIM_Base_Init(&htim8) != HAL_OK)
    {
        Error_Handler();
    }
    sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
    if (HAL_TIM_ConfigClockSource(&htim8, &sClockSourceConfig) != HAL_OK)
    {
        Error_Handler();
    }
    sMasterConfig.MasterOutputTrigger = TIM_TRGO_UPDATE;
    sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
    if (HAL_TIMEx_MasterConfigSynchronization(&htim8, &sMasterConfig) !=
HAL_OK)
    {
        Error_Handler();
    }
}
```

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

```
}
/* USER CODE BEGIN TIM8_Init 2 */

/* USER CODE END TIM8_Init 2 */

}

/**
 * @brief UART5 Initialization Function
 * @param None
 * @retval None
 */
static void MX_UART5_Init(void)
{
    /* USER CODE BEGIN UART5_Init 0 */

    /* USER CODE END UART5_Init 0 */

    /* USER CODE BEGIN UART5_Init 1 */

    /* USER CODE END UART5_Init 1 */
    huart5.Instance = UART5;
    huart5.Init.BaudRate = 9600;
    huart5.Init.WordLength = UART_WORDLENGTH_8B;
    huart5.Init.StopBits = UART_STOPBITS_1;
    huart5.Init.Parity = UART_PARITY_NONE;
    huart5.Init.Mode = UART_MODE_TX_RX;
    huart5.Init.HwFlowCtl = UART_HWCONTROL_NONE;
    huart5.Init.OverSampling = UART_OVERSAMPLING_16;
    if (HAL_UART_Init(&huart5) != HAL_OK)
    {
        Error_Handler();
    }
    /* USER CODE BEGIN UART5_Init 2 */

    /* USER CODE END UART5_Init 2 */

}

/**
 * @brief UART7 Initialization Function
 * @param None
 * @retval None
 */
static void MX_UART7_Init(void)
{
    /* USER CODE BEGIN UART7_Init 0 */

    /* USER CODE END UART7_Init 0 */

    /* USER CODE BEGIN UART7_Init 1 */

    /* USER CODE END UART7_Init 1 */

}
```

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

```
huart7.Instance = UART7;
huart7.Init.BaudRate = 9600;
huart7.Init.WordLength = UART_WORDLENGTH_8B;
huart7.Init.StopBits = UART_STOPBITS_1;
huart7.Init.Parity = UART_PARITY_NONE;
huart7.Init.Mode = UART_MODE_TX_RX;
huart7.Init.HwFlowCtl = UART_HWCONTROL_NONE;
huart7.Init.OverSampling = UART_OVERSAMPLING_16;
if (HAL_UART_Init(&huart7) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN UART7_Init 2 */
/* USER CODE END UART7_Init 2 */

}

/**
 * Enable DMA controller clock
 */
static void MX_DMA_Init(void)
{
    /* DMA controller clock enable */
    __HAL_RCC_DMA1_CLK_ENABLE();
    __HAL_RCC_DMA2_CLK_ENABLE();

    /* DMA interrupt init */
    /* DMA1_Stream0_IRQn interrupt configuration */
    HAL_NVIC_SetPriority(DMA1_Stream0_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(DMA1_Stream0_IRQn);
    /* DMA1_Stream1_IRQn interrupt configuration */
    HAL_NVIC_SetPriority(DMA1_Stream1_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(DMA1_Stream1_IRQn);
    /* DMA1_Stream3_IRQn interrupt configuration */
    HAL_NVIC_SetPriority(DMA1_Stream3_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(DMA1_Stream3_IRQn);
    /* DMA1_Stream5_IRQn interrupt configuration */
    HAL_NVIC_SetPriority(DMA1_Stream5_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(DMA1_Stream5_IRQn);
    /* DMA1_Stream6_IRQn interrupt configuration */
    HAL_NVIC_SetPriority(DMA1_Stream6_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(DMA1_Stream6_IRQn);
    /* DMA1_Stream7_IRQn interrupt configuration */
    HAL_NVIC_SetPriority(DMA1_Stream7_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(DMA1_Stream7_IRQn);
    /* DMA2_Stream2_IRQn interrupt configuration */
    HAL_NVIC_SetPriority(DMA2_Stream2_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(DMA2_Stream2_IRQn);
}

/**
 * @brief GPIO Initialization Function
 * @param None
 */
```

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

```
* @retval None
*/
static void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStruct = {0};

    /* GPIO Ports Clock Enable */
    __HAL_RCC_GPIOE_CLK_ENABLE();
    __HAL_RCC_GPIOC_CLK_ENABLE();
    __HAL_RCC_GPIOF_CLK_ENABLE();
    __HAL_RCC_GPIOH_CLK_ENABLE();
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOB_CLK_ENABLE();
    __HAL_RCC_GPIOG_CLK_ENABLE();
    __HAL_RCC_GPIOD_CLK_ENABLE();

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(GPIOE, GPIO_PIN_3, GPIO_PIN_RESET);

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(GPIOF, GPIO_PIN_8|GPIO_PIN_9|GPIO_PIN_10,
GPIO_PIN_RESET);

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(OTG_FS_PSO_GPIO_Port, OTG_FS_PSO_Pin, GPIO_PIN_RESET);

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(GPIOG, LD3_Pin|LD4_Pin, GPIO_PIN_RESET);

    /*Configure GPIO pins : PE2 PE4 PE5 PE6 */
    GPIO_InitStruct.Pin = GPIO_PIN_2|GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_6;
    GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    HAL_GPIO_Init(GPIOE, &GPIO_InitStruct);

    /*Configure GPIO pin : PE3 */
    GPIO_InitStruct.Pin = GPIO_PIN_3;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(GPIOE, &GPIO_InitStruct);

    /*Configure GPIO pins : PC13 PC14 PC15 */
    GPIO_InitStruct.Pin = GPIO_PIN_13|GPIO_PIN_14|GPIO_PIN_15;
    GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);

    /*Configure GPIO pins : PF0 PF1 PF2 PF3
    PF4 PF5 PF11 PF12
    PF13 PF14 PF15 */
    GPIO_InitStruct.Pin = GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_2|GPIO_PIN_3
|GPIO_PIN_4|GPIO_PIN_5|GPIO_PIN_11|GPIO_PIN_12
|GPIO_PIN_13|GPIO_PIN_14|GPIO_PIN_15;
    GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
```

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

```
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
GPIO_InitStruct.Alternate = GPIO_AF12_FMC;
HAL_GPIO_Init(GPIOF, &GPIO_InitStruct);

/*Configure GPIO pins : PF8 PF9 PF10 */
GPIO_InitStruct.Pin = GPIO_PIN_8|GPIO_PIN_9|GPIO_PIN_10;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(GPIOF, &GPIO_InitStruct);

/*Configure GPIO pin : PC0 */
GPIO_InitStruct.Pin = GPIO_PIN_0;
GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
GPIO_InitStruct.Alternate = GPIO_AF12_FMC;
HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);

/*Configure GPIO pin : B1_Pin */
GPIO_InitStruct.Pin = B1_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_EVT_RISING;
GPIO_InitStruct.Pull = GPIO_NOPULL;
HAL_GPIO_Init(B1_GPIO_Port, &GPIO_InitStruct);

/*Configure GPIO pin : BOOT1_Pin */
GPIO_InitStruct.Pin = BOOT1_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
GPIO_InitStruct.Pull = GPIO_NOPULL;
HAL_GPIO_Init(BOOT1_GPIO_Port, &GPIO_InitStruct);

/*Configure GPIO pins : PG0 PG1 PG4 PG5
                        PG8 PG15 */
GPIO_InitStruct.Pin = GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_4|GPIO_PIN_5
                    |GPIO_PIN_8|GPIO_PIN_15;
GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
GPIO_InitStruct.Alternate = GPIO_AF12_FMC;
HAL_GPIO_Init(GPIOG, &GPIO_InitStruct);

/*Configure GPIO pins : PE7 PE8 PE9 PE10
                        PE11 PE12 PE13 PE14
                        PE15 PE0 PE1 */
GPIO_InitStruct.Pin = GPIO_PIN_7|GPIO_PIN_8|GPIO_PIN_9|GPIO_PIN_10
                    |GPIO_PIN_11|GPIO_PIN_12|GPIO_PIN_13|GPIO_PIN_14
                    |GPIO_PIN_15|GPIO_PIN_0|GPIO_PIN_1;
GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
GPIO_InitStruct.Alternate = GPIO_AF12_FMC;
HAL_GPIO_Init(GPIOE, &GPIO_InitStruct);

/*Configure GPIO pins : OTG_HS_ID_Pin OTG_HS_DM_Pin OTG_HS_DP_Pin */
```



## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

```
GPIO_InitStruct.Pin = OTG_HS_ID_Pin|OTG_HS_DM_Pin|OTG_HS_DP_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
GPIO_InitStruct.Alternate = GPIO_AF12_OTG_HS_FS;
HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

/*Configure GPIO pin : VBUS_HS_Pin */
GPIO_InitStruct.Pin = VBUS_HS_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
GPIO_InitStruct.Pull = GPIO_NOPULL;
HAL_GPIO_Init(VBUS_HS_GPIO_Port, &GPIO_InitStruct);

/*Configure GPIO pins : PD8 PD9 PD10 PD14
                        PD15 PD0 PD1 */
GPIO_InitStruct.Pin = GPIO_PIN_8|GPIO_PIN_9|GPIO_PIN_10|GPIO_PIN_14
                    |GPIO_PIN_15|GPIO_PIN_0|GPIO_PIN_1;
GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
GPIO_InitStruct.Alternate = GPIO_AF12_FMC;
HAL_GPIO_Init(GPIOD, &GPIO_InitStruct);

/*Configure GPIO pin : OTG_FS_OC_Pin */
GPIO_InitStruct.Pin = OTG_FS_OC_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING;
GPIO_InitStruct.Pull = GPIO_NOPULL;
HAL_GPIO_Init(OTG_FS_OC_GPIO_Port, &GPIO_InitStruct);

/*Configure GPIO pin : OTG_FS_PSO_Pin */
GPIO_InitStruct.Pin = OTG_FS_PSO_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(OTG_FS_PSO_GPIO_Port, &GPIO_InitStruct);

/*Configure GPIO pins : STLINK_RX_Pin STLINK_TX_Pin */
GPIO_InitStruct.Pin = STLINK_RX_Pin|STLINK_TX_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
GPIO_InitStruct.Alternate = GPIO_AF7_USART1;
HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

/*Configure GPIO pins : LD3_Pin LD4_Pin */
GPIO_InitStruct.Pin = LD3_Pin|LD4_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(GPIOG, &GPIO_InitStruct);

/*Configure GPIO pins : PB5 PB6 */
GPIO_InitStruct.Pin = GPIO_PIN_5|GPIO_PIN_6;
GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
```

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

```
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;
GPIO_InitStruct.Alternate = GPIO_AF12_FMC;
HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

}

/* USER CODE BEGIN 4 */
/* ----- CALLBACKS ----- */

void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc)
{
//Conversión de bits a Voltios reales (V)
//V4
volts[0]= (((float)valor_adc[0] / RESOLUCION) * MAX_VOLTS_V4)*GANANCIA_V4;
//SENSOR DE CORRIENTE DRON
volts[1]= (((float)valor_adc[1] / RESOLUCION) * MAX_VOLTS_DRON;
//SENSOR DE CORRIENTE PILA
volts[2]= (((float)valor_adc[2] / RESOLUCION) * MAX_VOLTS_PILA;
//Presión (P)
volts[3]= (((float)valor_adc[3] / RESOLUCION) * MAX_VOLTS_ADC;
//Tensión de salida de la pila (Vs)
volts[4]= (((float)valor_adc[4] / RESOLUCION) * MAX_VOLTS_ADC)*GANANCIA_VS;
//Temperatura de la pila (T)
volts[5]= (((float)valor_adc[5] / RESOLUCION) * MAX_VOLTS_ADC;
//Tensión de celda (Vp)
volts[6]= (((float)valor_adc[6] / RESOLUCION) * MAX_VOLTS_ADC)*GANANCIA_VP;
//V1
volts[7]= (((float)valor_adc[7] / RESOLUCION) * MAX_VOLTS_V1)*GANANCIA_V1;
//V2
volts[8]= (((float)valor_adc[8] / RESOLUCION) * MAX_VOLTS_V2)*GANANCIA_V2;
//V3
volts[9]= (((float)valor_adc[9] / RESOLUCION) * MAX_VOLTS_V3)*GANANCIA_V3;
//Flujo de oxígeno (O2)
volts[10]= (((float)valor_adc[10] / RESOLUCION) * MAX_VOLTS_ADC;

// Las señales de los sensores de corriente se cambian de nombre y se pasan a
Amperios reales (2.7V -> 3 A)
corriente[0]=volts[1]*GANANCIA_VDRON; // Im
corriente[1]=volts[2]*GANANCIA_VPILA; // I
// Las señal del nivel de presión se cambia de nombre y se pasa a unidades de
presión (2.7 V -> 1 bar)
presion=volts[3]*GANANCIA_PRESION; // P
// Las señal de la concentración de oxígeno se cambia de nombre y se pasa a
unidades de concentración (2.7 V -> 21 % O2)
concentracion_O2=volts[10]; // O2
// Las señal de la temperatura se cambia de nombre y se pasa a unidades de
temperatura (2.7 V -> 70 °C)
temperatura=volts[5]*GANANCIA_T; // T
//Indicador visual de conversión finalizada (LED3 ON/LED3 OFF)
HAL_GPIO_TogglePin(GPIOD, LD3_Pin);

}

/* ----- General Purpose Functions ----- */
```

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

```
/* ----- Guard Functions ----- */
// H2DRONE
static uint8_t bat_desequilibrada_or_max (fsm_t* this){
// Se activa si la batería está en carga y alguna de las tensiones de celda
está desequilibrada
if ((corriente[1]>corriente[0]) && (volts[0]>MAX_TENSION_CARGA)){
return(1);
} else if (volts[7]>((volts[0]/CELDAS_BAT)+(PERDIDAS_BAT*volts[0]))
|| volts[7]<((volts[0]/CELDAS_BAT)-
(PERDIDAS_BAT*volts[0])))return(1);
} else if (volts[8]>((volts[0]/CELDAS_BAT)+(PERDIDAS_BAT*volts[0]))
|| volts[8]<((volts[0]/CELDAS_BAT)-(PERDIDAS_BAT*volts[0]))){
return(1);
} else if (volts[9]>((volts[0]/CELDAS_BAT)+(PERDIDAS_BAT*volts[0]))
volts[9]<((volts[0]/CELDAS_BAT)-(PERDIDAS_BAT*volts[0]))){
return(1);
} else if (volts[0]>((volts[0]/CELDAS_BAT)+(PERDIDAS_BAT*volts[0]))
|| volts[0]<((volts[0]/CELDAS_BAT)-(PERDIDAS_BAT*volts[0]))){
return(1);
}
else {
return(0);
}
}

static uint8_t bat_equilibrada (fsm_t* this){
//Se activa si la batería está en carga y todas las tensiones de celda
están equilibradas
if ((volts[3]>volts[2])
&&(volts[0]<MAX_TENSION_CARGA)
&& ((volts[0]/CELDAS_BAT)+(PERDIDAS_BAT*volts[0]))>volts[7]
&& volts[7]>((volts[0]/CELDAS_BAT)-(PERDIDAS_BAT*volts[0]))
&& ((volts[0]/CELDAS_BAT)+(PERDIDAS_BAT*volts[0]))>volts[8]
&& volts[8]>((volts[0]/CELDAS_BAT)-(PERDIDAS_BAT*volts[0]))
&& ((volts[0]/CELDAS_BAT)+(PERDIDAS_BAT*volts[0]))>volts[9]
&& volts[9]>((volts[0]/CELDAS_BAT)-(PERDIDAS_BAT*volts[0]))
&& ((volts[0]/CELDAS_BAT)+(PERDIDAS_BAT*volts[0]))>volts[0]
&& volts[0]>((volts[0]/CELDAS_BAT)-(PERDIDAS_BAT*volts[0]))){
return(1);
}
else {
return(0);
}
}

static uint8_t temperatura_alta (fsm_t* this){
//Se activa cuando la temperatura es igual o supera los 40 °C hasta
los 65 °C
if((TEMP_MAX >= temperatura) && (temperatura >= TEMP_MIN)){
return(1);
}
else{
return(0);
}
}
```

```
    }
}

static uint8_t temperatura_baja (fsm_t* this){
    //Se activa cuando la temperatura es inferior a 40 °C
    if(temperatura<TEMP_MIN){
        return(1);
    }
    else{
        return(0);
    }
}

static uint8_t tension_de_celda_baja (fsm_t* this){
    //Cálculo de la tensión mínima de celda
    float tension_de_celda_min=(volts[4]/NUMERO_CELDAS) -
    PERDIDAS_PILA*(volts[4]/NUMERO_CELDAS);

    //Se activa si la tensión de celda disminuye un 10%
    if(volts[6]<tension_de_celda_min){
        return(1);
    }
    else {
        return(0);
    }
}

static uint8_t tiempo_apertura_C4 (fsm_t* this){
    //Se activa una vez acaba el tiempo del temporizador virtual (500 ms)
    if(timer_ended(&timer_C4)==1){
        //La flag se actualiza antes de ejecutarse la FSM
        return (1);
    }
    else{
        return(0);
    }
}

static uint8_t parametros_de_seguridad (fsm_t* this){
    //Se activa si P, Vs, T o I se sale de su rango de seguridad
    if (presion>=PRESION_MAX){
        return(1);
    }
    if (corriente[1]>=I_MAX){
        return(1);
    }
    if (volts[4]<VS_MIN || volts[4]>VS_MAX){
        return(1);
    }
    if (temperatura>TEMP_MAX){
        return(1);
    }
    else{
        return(0);
    }
}
}
```

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

```
    }
}

static uint8_t parametros_rango_normal (fsm_t* this){
    //Se activa si P, Vs, T y I están dentro de su rango de trabajo normal
    if ((presion<PRESION_MAX)
        &&(corriente[1]<I_MAX)
        &&(volts[4]>=VS_MIN)
        &&(volts[4]<=VS_MAX)
        &&(temperatura<=TEMP_MAX)){
        return(1);
    }
    else {
        return(0);
    }
}

/* ----- Action Functions ----- */
// H2DRONE
static void corriente_de_consumo (fsm_t* this){
    // Convertir corriente a tensión proporcional (3.0 es la máxima
    tensión del DAC)
    volatile float volts_corriente_consumo =
    (corriente[0]/I_MAX)*VDAC_MAX;

    // Convertir el valor de tensión a la escala del DAC (0 a 4095)
    dac_value2=
    (uint32_t)(volts_corriente_consumo*DIV_RESOLUCION_VDACMAX);
}

static void carga_normal (fsm_t* this){
    // Sumar 1 A a la corriente de consumo
    volatile float corriente_carga_normal = corriente[0]+1.0;

    // Convertir corriente a tensión proporcional (3.0 es la máxima
    tensión del DAC)
    volatile float volts_carga_normal =
    (corriente_carga_normal/I_MAX)*VDAC_MAX;

    // Convertir el valor de tensión a la escala del DAC (0 a 4095)
    dac_value2 = (uint32_t)(volts_carga_normal*DIV_RESOLUCION_VDACMAX);
}

static void ventilador_carga_media (fsm_t* this){
    // Ventilador al 50%
    dac_value1= 2047; // Buffer con el valor medio del rango del DAC
}

static void ventilador_proporcional_I_Q (fsm_t* this){
    // Ventilador proporcional a la corriente de salida de la pila y a la
    variación de caudal de aire.

    //Fórmula variación del caudal de aire
    volatile float variacion_caudal = 0.21/concentracion_O2;
```

```
// Cálculo de la tensión proporcional a la corriente
volatile float volts_prop_I_Q = (corriente[1]/I_MAX)*VDAC_MAX;

//Variación de caudal afecta a la velocidad del ventilador
if (variacion_caudal<0.0777){
/*Con un 21% de oxígeno la variación del caudal es la mínima
(0.0777),si es mayor, hay que aumentar velocidad del ventilador */
    volts_prop_I_Q=volts_prop_I_Q*variacion_caudal;
}

// Convertir el valor de tensión a la escala del DAC (0 a 4095)
dac_value1 = (uint32_t)(volts_prop_I_Q * DIV_RESOLUCION_VDACMAX);
}

static void ventilador_prporcional_T_Q (fsm_t* this){
// Ventilador proporcional a la temperatura de la pila y a la
variación de caudal de aire.
volatile float variacion_caudal = 0.21/concentracion_O2; //Fórmula
variacion del caudal de aire

// Cálculo de la tensión proporcional
volatile float volts_prop_T_Q = (temperatura/TEMP_MAX)*VDAC_MAX;

//Variación de caudal afecta a la velocidad del ventilador
if (variacion_caudal<0.0777){
    volts_prop_T_Q=volts_prop_T_Q*variacion_caudal;
}

// Convertir el valor de tensión a la escala del DAC (0 a 4095)
dac_value1 = (uint32_t)(volts_prop_T_Q * DIV_RESOLUCION_VDACMAX);
}

static void abrir_C4 (fsm_t* this){
HAL_GPIO_WritePin(GPIOF, GPIO_PIN_10, 1); //Abrir válvula de purgado
(C4)
timer_set(&timer_C4, TIEMPO_APERTURA_C4); //Set timer virtual para
abrir válvula C4 durante 500 ms.
}

static void cerrar_C4 (fsm_t* this){
HAL_GPIO_WritePin(GPIOF, GPIO_PIN_10, 0); //Cerrar válvula de purgado
(C4)
timer_disable(&timer_C4); //Deshabilitar timer virtual
}

static void abrir_circuito (fsm_t* this){
HAL_GPIO_WritePin(GPIOF, GPIO_PIN_8, 0); //Cerrar válvula principal
(C1)
HAL_GPIO_WritePin(GPIOF, GPIO_PIN_9, 1); //Abrir relé de seguridad
(C5)
}

static void cerrar_circuito (fsm_t* this){
HAL_GPIO_WritePin(GPIOF, GPIO_PIN_8, 1); //Abrir válvula principal
(C1)
```

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

```
        HAL_GPIO_WritePin(GPIOF, GPIO_PIN_9, 0); //Cerrar relé de seguridad
        (C5)
    }

/* USER CODE END 4 */

/**
 * @brief Period elapsed callback in non blocking mode
 * @note This function is called when TIM6 interrupt took place, inside
 * HAL_TIM_IRQHandler(). It makes a direct call to HAL_IncTick() to
increment
 * a global variable "uwTick" used as application time base.
 * @param htim : TIM handle
 * @retval None
 */
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
    /* USER CODE BEGIN Callback 0 */
    /* USER CODE END Callback 0 */
    /* USER CODE BEGIN Callback 1 */

    /* USER CODE END Callback 1 */
}

/**
 * @brief This function is executed in case of error occurrence.
 * @retval None
 */
void Error_Handler(void)
{
    /* USER CODE BEGIN Error_Handler_Debug */
    /* User can add his own implementation to report the HAL error return state
 */
    __disable_irq();
    while (1)
    {
    }
    /* USER CODE END Error_Handler_Debug */
}

#ifdef USE_FULL_ASSERT
/**
 * @brief Reports the name of the source file and the source line number
 * where the assert_param error has occurred.
 * @param file: pointer to the source file name
 * @param line: assert_param error line source number
 * @retval None
 */
void assert_failed(uint8_t *file, uint32_t line)
{
    /* USER CODE BEGIN 6 */
    /* User can add his own implementation to report the file name and line
number,
ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line)
 */

```

# Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

```
/* USER CODE END 6 */  
}  
#endif /* USE_FULL_ASSERT */
```

## 6.2 main.h

```
/* USER CODE BEGIN Header */  
/**  
  
*****  
*  
* @file : main.h  
* @brief : Header for main.c file.  
* This file contains the common defines of the  
application.  
  
*****  
* @attention  
*  
* Copyright (c) 2023 STMicroelectronics.  
* All rights reserved.  
*  
* This software is licensed under terms that can be found in the LICENSE  
file  
* in the root directory of this software component.  
* If no LICENSE file comes with this software, it is provided AS-IS.  
*  
  
*****  
*/  
/* USER CODE END Header */  
  
/* Define to prevent recursive inclusion -----  
-*/  
#ifndef __MAIN_H  
#define __MAIN_H  
  
#ifdef __cplusplus  
extern "C" {  
#endif  
  
/* Includes -----*/  
#include "stm32f4xx_hal.h"  
  
/* Private includes -----*/  
/* USER CODE BEGIN Includes */  
  
/* USER CODE END Includes */  
/* Exported types -----*/  
/* USER CODE BEGIN ET */  
  
/* USER CODE END ET */  
  
/* Exported constants -----*/
```



## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

```
/* USER CODE BEGIN EC */

/* USER CODE END EC */

/* Exported macro -----*/
/* USER CODE BEGIN EM */

/* USER CODE END EM */

/* Exported functions prototypes -----*/
void Error_Handler(void);

/* USER CODE BEGIN EFP */

/* USER CODE END EFP */

/* Private defines -----*/
#define PH0_OSC_IN_Pin GPIO_PIN_0
#define PH0_OSC_IN_GPIO_Port GPIOH
#define PH1_OSC_OUT_Pin GPIO_PIN_1
#define PH1_OSC_OUT_GPIO_Port GPIOH
#define B1_Pin GPIO_PIN_0
#define B1_GPIO_Port GPIOA
#define BOOT1_Pin GPIO_PIN_2
#define BOOT1_GPIO_Port GPIOB
#define OTG_HS_ID_Pin GPIO_PIN_12
#define OTG_HS_ID_GPIO_Port GPIOB
#define VBUS_HS_Pin GPIO_PIN_13
#define VBUS_HS_GPIO_Port GPIOB
#define OTG_HS_DM_Pin GPIO_PIN_14
#define OTG_HS_DM_GPIO_Port GPIOB
#define OTG_HS_DP_Pin GPIO_PIN_15
#define OTG_HS_DP_GPIO_Port GPIOB
#define OTG_FS_OC_Pin GPIO_PIN_12
#define OTG_FS_OC_GPIO_Port GPIOD
#define OTG_FS_PSO_Pin GPIO_PIN_13
#define OTG_FS_PSO_GPIO_Port GPIOD
#define STLINK_RX_Pin GPIO_PIN_9
#define STLINK_RX_GPIO_Port GPIOA
#define STLINK_TX_Pin GPIO_PIN_10
#define STLINK_TX_GPIO_Port GPIOA
#define SWDIO_Pin GPIO_PIN_13
#define SWDIO_GPIO_Port GPIOA
#define SWCLK_Pin GPIO_PIN_14
#define SWCLK_GPIO_Port GPIOA
#define LD3_Pin GPIO_PIN_13
#define LD3_GPIO_Port GPIOG
#define LD4_Pin GPIO_PIN_14
#define LD4_GPIO_Port GPIOG

/* USER CODE BEGIN Private defines */
// N° DATOS PARA HACER GRÁFICAS CON MATLAB
#define DATOS_GRAFICAS 11
```

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

```
//NUMERO DE CELDAS DE LA PILA DE COMBUSTIBLE
#define NUMERO_CELDAS 20

//NUMERO DE CANALES ADC
#define CANALES_ADC 12

//RESOLUCION ADC
#define RESOLUCION 4095.00

//NUMERO DE SENSORES DE CORRIENTE
#define SENSORES_DE_CORRIENTE 2

//TIEMPO DE EJECUCION FSM
#define T_FIRE 2

//TIEMPO DE APERTURA DE LA VÁLVULA DE PURGADO C4
#define TIEMPO_APERTURA_C4 500

//TENSIONES MAXIMAS
//MAXIMA TENSIÓN DE CARGA DE LA BATERÍA
#define MAX_TENSION_CARGA 16.59

//MAXIMO POCENTAJE PERMITIDO DE PÉRDIDAS EN LAS CELDAS DE LA BATERÍA
#define PERDIDAS_BAT 0.1

//NÚMERO DE CELDAS DE LA BATERÍA LIPO
#define CELDAS_BAT 4.00

//MAXIMO POCENTAJE PERMITIDO DE PÉRDIDAS EN LAS CELDAS DE LA PILA
#define PERDIDAS_PILA 0.1

//CORRIENTE MÁXIMA DE SALIDA DE LA PILA
#define I_MAX 3.00

//TEMPERATURA DE LA PILA MÍNIMA Y MÁXIMA
#define TEMP_MAX 65.0
#define TEMP_MIN 40.0

//PRESIÓN MÁXIMA DEL DEPÓSITO DE H2 DE LA PILA
#define PRESION_MAX 0.6

//TENSION DE SALIDA DE LA PILA MÍNIMA Y MÁXIMA
#define VS_MAX 24.0
#define VS_MIN 10.0

//TENSIONES MÁXIMAS DE ENTRADA AL ADC
#define MAX_VOLTS_ADC 2.70
#define MAX_VOLTS_V1 2.707
#define MAX_VOLTS_V2 2.665
#define MAX_VOLTS_V3 2.702
#define MAX_VOLTS_V4 2.697
#define MAX_VOLTS_DRON 2.866
#define MAX_VOLTS_PILA 2.838
```

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

```
//GANANCIAS PARA LA CONVERSIÓN DE UNIDADES
```

```
#define GANANCIA_V1 2.51611  
#define GANANCIA_V2 4.2901  
#define GANANCIA_V3 5.937  
#define GANANCIA_V4 7.65  
#define GANANCIA_VDRON 1.44  
#define GANANCIA_VPILA 1.4  
#define GANANCIA_PRESION 0.370  
#define GANANCIA_VS 9.259  
#define GANANCIA_T 25.925  
#define GANANCIA_VP 0.4629  
#define GANANCIA_O2 0.078
```

```
//DAC
```

```
#define VDAC_MAX 3.0  
#define DIV_RESOLUCION_VDACMAX 1365
```

```
/* USER CODE END Private defines */
```

```
#ifndef __cplusplus
```

```
}
```

```
#endif
```

```
#endif /* __MAIN_H */
```





UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



# **ANEXO II**

**Manual de usuario de hardware**

**Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”**



## **ÍNDICE DEL ANEXO II**

<b>1. INTRODUCCIÓN</b> .....	1
<b>2. ESQUEMA DEL SISTEMA</b> .....	2
<b>3. BATERÍA LIPO</b> .....	3
<b>4. CARGA ELECTRÓNICA</b> .....	4
<b>5. FUENTE DE ALIMENTACIÓN (PILA DE COMBUSTIBLE)</b> .....	4
<b>6. PLACA DE INTERFAZ DE HARDWARE</b> .....	6





## **ÍNDICE DE FIGURAS DEL ANEXO II**

<b>Figura 1.</b> Esquema de las conexiones del sistema. ....	2
<b>Figura 2.</b> Conexión de la batería LiPo. ....	3
<b>Figura 3.</b> Conexión sensores de corriente a la fuente de alimentación que simula la pila de combustible. ....	5
<b>Figura 4.</b> Conexión del microcontrolador a la placa de interfaz de hardware. ....	6
<b>Figura 5.</b> Conexiones regleta. ....	7



## **1. INTRODUCCIÓN**

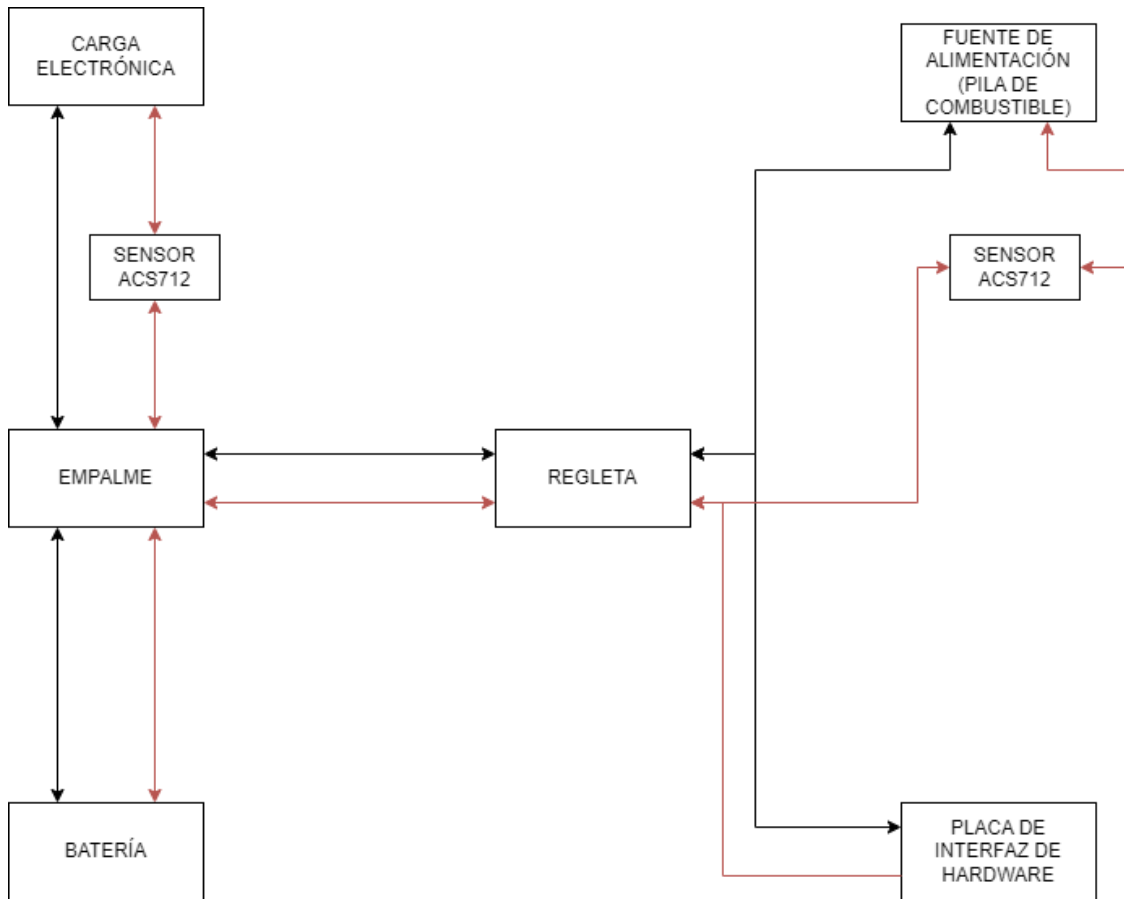
A continuación, se ha hecho una pequeña guía con los pasos que hay que seguir para poner el sistema en funcionamiento. En este caso se consideran los pasos para poner en marcha la carga electrónica, la placa de desarrollo y la placa de interfaz de hardware. Esto no quiere decir que lleva el sistema real con un dron se arme de diferente manera.

Previo al comienzo de la guía, se recomienda leer las siguientes indicaciones para un correcto uso del sistema:

- Antes de manipular la placa PCB, la carga electrónica o de conectar cualquier sensor o microcontrolador, asegúrate de que la alimentación (batería LiPo) esté desconectada para evitar cortocircuitos y daños a los componentes.
- Verifica que el voltaje de alimentación sea el adecuado para la placa PCB. No excedas los límites de voltaje especificados, ya que podría resultar en daños irreparables.
- Evita tocar los componentes electrónicos con las manos desnudas, ya que la electricidad estática puede dañarlos. Utiliza una pulsera antiestática o toca una superficie metálica a tierra antes de manipular la PCB.
- Si la placa PCB genera calor considerable, asegúrate de proporcionar una adecuada disipación de calor mediante el uso de disipadores de calor, ventiladores o elementos de enfriamiento adicionales, según sea necesario.
- Si es necesario limpiar la placa PCB, utiliza productos y herramientas adecuadas que no dañen los componentes ni los trazos de cobre. Evita el uso de solventes agresivos o abrasivos.

## 2. ESQUEMA DEL SISTEMA

El esquema de las conexiones del sistema, donde los cables rojos representan los que se conectan al terminal positivo, y los negros al negativo, se muestra en la Figura 1:



**Figura 48. Esquema de las conexiones del sistema.**

Cabe destacar, que previamente a la conexión de dispositivos en el sistema, se debe realizar una calibración de los potenciómetros de la placa de interfaz de hardware. Esto incluye los potenciómetros de los divisores de tensión y los potenciómetros que proporcionan ganancia a los Amplificadores de Instrumentación AD623.

La calibración consiste en introducir el voltaje máximo correspondiente en cada una de las entradas, y girar el potenciómetro con un destornillador de punta fina con el fin de obtener 2.7 V o un valor bastante aproximado en la salida. Estos valores se deben de introducir en el apartado de “#defines”, como se menciona en el Anexo II, con el fin de optimizar los cálculos.

### 3. BATERÍA LIPO

- Antes de comenzar, verifica que la batería esté cargada y en un estado óptimo. Consulta las indicaciones del fabricante sobre cómo verificar el nivel de carga.
- Identifica la polaridad correcta de los conectores banana (rojo para el positivo y negro para el negativo) y verifica que coincida con los puertos o terminales de destino.
- Al igual que con los conectores banana, asegúrate de identificar la polaridad correcta de los conectores JST y verifica que coincida con los puertos o terminales de destino.

La conexión de la batería se hará después de la conexión del resto de componentes, por seguridad, y se hará por dos de los conectores macho del empalme, como se observa en la Figura 2:

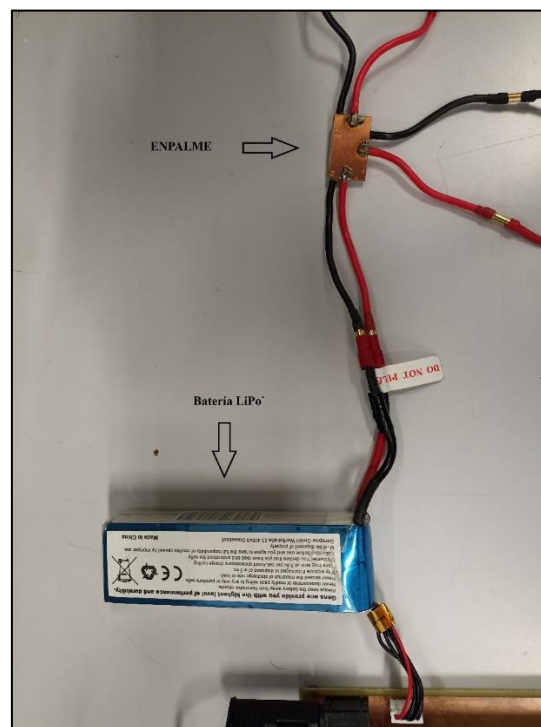


Figura 49. Conexión de la batería LiPo.

#### **4. CARGA ELECTRÓNICA**

El consumo de corriente de la carga electrónica se controla a través de un programa de comunicación serie, a través del cual se le envían comandos para aumentar o disminuir el consumo. Esta carga, necesita de la tensión de red para su alimentación, y de una tensión DC de cualquier valor para poder consumir corriente.

El programa de comunicación serie recomendado es PuTTY, al cual se le introduce el puerto correspondiente, y la velocidad de comunicación de baudios (9600).

La carga electrónica manda constantemente el valor de la tensión que está recibiendo, la corriente que consume, y de su temperatura interna. Para comenzar a consumir corriente, el usuario debe mandar la letra “C” seguida de un número entre 0 y 4, el cual corresponde al valor de intensidad que el usuario quiere que se consuma.

Ejemplo:

C1 → 1 Amperio

C2 → 2 Amperios

C3.5 → 3.5 Amperios

El terminal positivo de la carga electrónica se debe conectar al terminal negativo de un sensor de corriente ACS712 a través de un conector banana macho – banana macho 3.5 mm, y el terminal positivo del sensor al empalme a través de uno de sus conectores hembra.

El terminal negativo de la carga se conecta al empalme con el otro conector hembra a través un cable banana macho – banana macho 3.5 mm.

El sensor conectado es el encargado de la medición de la corriente de consumo del dron (carga electrónica en este caso).

#### **5. FUENTE DE ALIMENTACIÓN (PILA DE COMBUSTIBLE)**

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

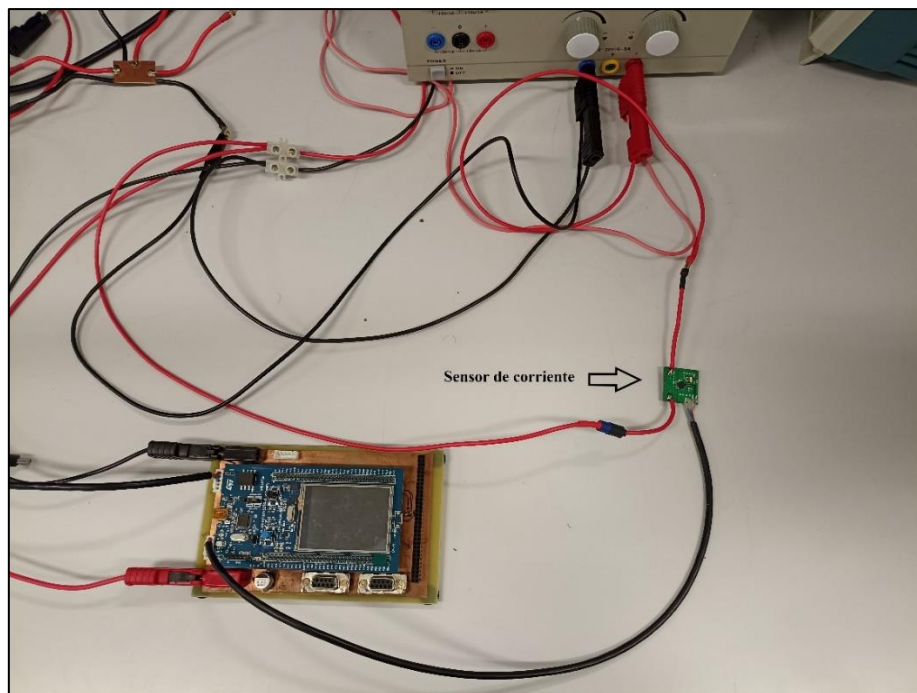
La fuente de alimentación es la encargada de proporcionar las tensiones que entregaría la pila de combustible real.

Se debe conectar el terminal positivo del otro sensor de corriente al terminal positivo de la fuente, a través de un conector banana macho – banana hembra 3.5 mm, y el terminal negativo del sensor a la regleta a través de un cable desnudo – banana macho 3.5 mm.

El terminal negativo de la fuente va conectado directamente a la regleta a través de un cable desnudo - banana macho.

El sensor de corriente conectado a la fuente de alimentación es el encargado de la medición de su corriente de salida, la cual simula la corriente de salida de la pila de combustible.

Las conexiones se observan en la siguiente Figura 3:

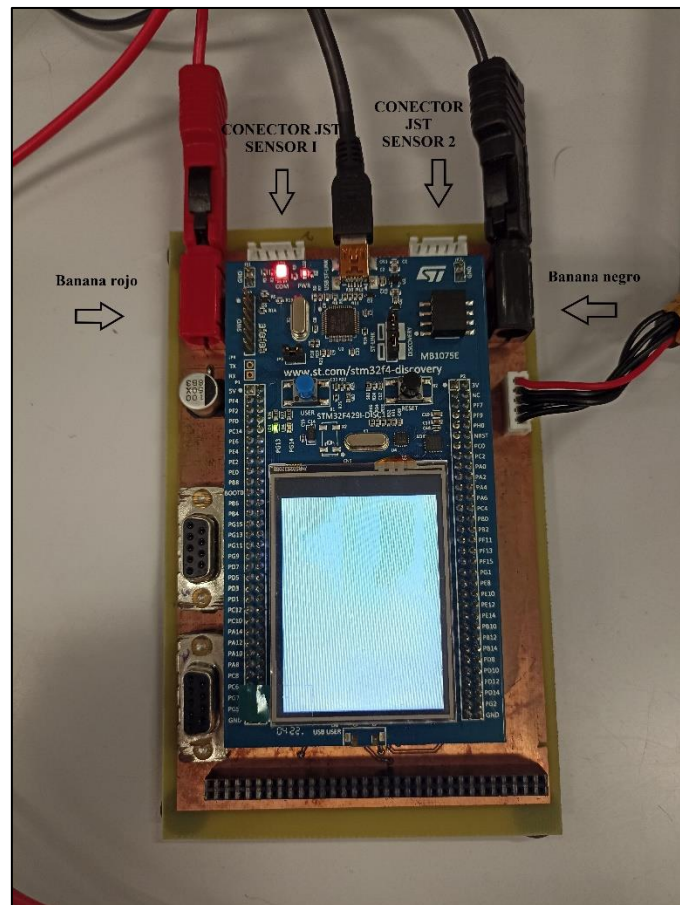


**Figura 50. Conexión sensores de corriente a la fuente de alimentación que simula la pila de combustible.**

## **6. PLACA DE INTERFAZ DE HARDWARE**

La placa de interfaz de hardware está compuesta por tres conectores GPIO, tres conectores JST hembra, dos conectores banana hembra (rojo / negro) y dos conectores DB-9 hembra.

La placa de desarrollo STM32F429I-DISC1 debe conectarse los dos conectores GPIO que se encuentran en paralelo, asegurándose de que está en el sentido correcto, como se muestra en la siguiente Figura 4:



**Figura 51. Conexión del microcontrolador a la placa de interfaz de hardware.**

Previamente a la conexión de la placa de desarrollo, se debe de subir el programa de control al microcontrolador, el cual se muestra en el Anexo I, siguiendo los pasos del Anexo III.

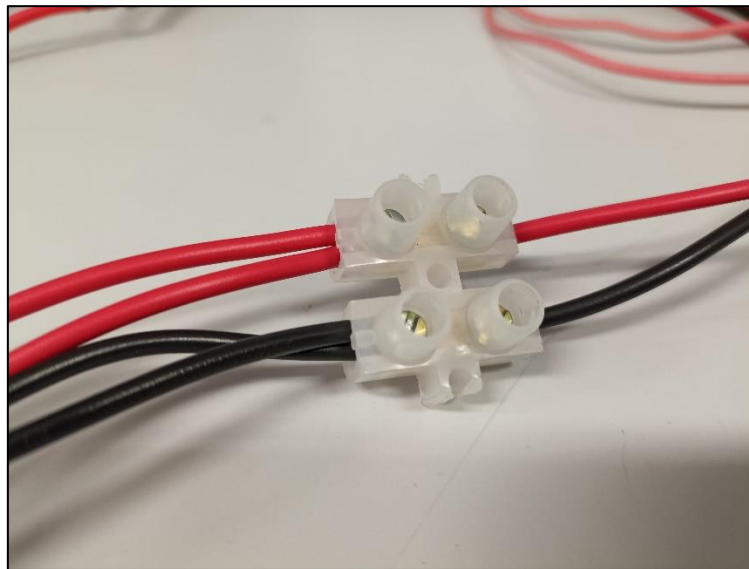


## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

Los cables banana hembra se deben conectar a la regleta por el mismo lado que la fuente de alimentación para crear una conexión en paralelo. Esta conexión se debe realizar mediante dos cables desnudos - banana macho, asegurándose de conectar en la regleta el negativo de la placa (negro) al negativo de la fuente, y el positivo de la placa (rojo) al positivo de la fuente.

La conexión que debe resultar se muestra en la Figura 5:



**Figura 52. Conexiones regleta.**

Los cables de alimentación (rojo), transmisión de señal (verde) y masa (azul) de los sensores de corriente ACS712 deben conectarse a los dos conectores JST hembra que se encuentran entre los conectores banana hembra rojo y negro. En el conector JST hembra restante se debe conectar el conector JST macho de la batería LiPo (tensiones de celda).

Por último, los conectores DB-9 hembra se conectan al ordenador a través de un cable DB9 macho - DB9 macho y un adaptador a USB, y permiten establecer la comunicación serie.





UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



# **ANEXO III**

**Manual de usuario de software**

**Implementación de Máquina de  
Estados Finita en Microcontrolador  
para optimizar el control del sistema  
de energía de un dron híbrido “pila  
de combustible – batería”**



## **ÍNDICE DEL ANEXO III**

<b>1. INTRODUCCIÓN.....</b>	<b>1</b>
<b>2. SELECCIÓN DE TARJETA .....</b>	<b>2</b>
<b>3. CONFIGURACIÓN DE PERIFÉRICOS .....</b>	<b>3</b>
<b>4. CÓDIGO Y MODO RUN .....</b>	<b>5</b>



## **ÍNDICE DE FIGURAS DEL ANEXO III**

<b>Figura 1.</b> Selección de tarjetas. ....	2
<b>Figura 2.</b> Configuración ADC. ....	3
<b>Figura 3.</b> Configuración UART. ....	4
<b>Figura 4.</b> main.h del proyecto. ....	5
<b>Figura 5.</b> Modo RUN. ....	6





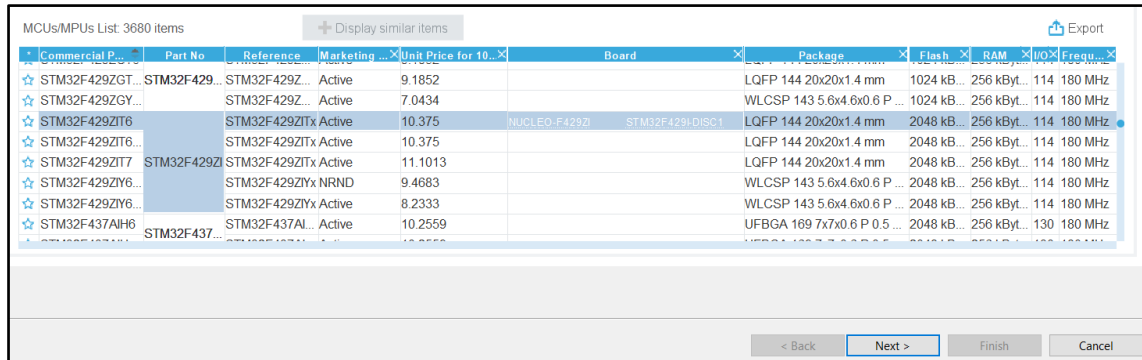
## **1. INTRODUCCIÓN**

A continuación, se ha hecho una pequeña guía con los pasos que hay que seguir para poner el software en marcha. En este caso se consideran los pasos para configurar los periféricos y ajustar el código fuente para un control optimizado.

Previamente, se debe disponer de un ordenador con entrada USB, disponer de la placa de desarrollo STM32F429I-DISC1 junto con un cable USB 2.0, y del entorno de desarrollo STM32CubeIDE.

## 2. SELECCIÓN DE TARJETA

El primer paso es crear un proyecto en STM32CubeIDE y seleccionar la tarjeta que estamos empleando. En nuestro caso es STM32F429ZIT6, como se muestra en la Figura 1:



MCUs/MPUs List: 3680 items

Commercial P...	Part No.	Reference	Marketing ...	Unit Price for 10...	Board	Package	Flash	RAM	I/O	Frequ...
☆	STM32F429ZGT...	STM32F429...	STM32F429Z...	Active	9.1852		LQFP 144 20x20x1.4 mm	1024 kB...	256 kByt...	114 180 MHz
☆	STM32F429ZGY...	STM32F429...	STM32F429Z...	Active	7.0434		WLCSP 143 5.6x4.6x0.6 P...	1024 kB...	256 kByt...	114 180 MHz
☆	STM32F429ZIT6	STM32F429ZITx	Active	10.375	NUCLEO-F429Z	STM32F429I-DISC1	LQFP 144 20x20x1.4 mm	2048 kB...	256 kByt...	114 180 MHz
☆	STM32F429ZIT7	STM32F429ZITx	Active	10.375			LQFP 144 20x20x1.4 mm	2048 kB...	256 kByt...	114 180 MHz
☆	STM32F429ZIT7	STM32F429ZITx	Active	11.1013			LQFP 144 20x20x1.4 mm	2048 kB...	256 kByt...	114 180 MHz
☆	STM32F429ZIY6...	STM32F429ZIYx	NRND	9.4683			WLCSP 143 5.6x4.6x0.6 P...	2048 kB...	256 kByt...	114 180 MHz
☆	STM32F429ZIY6...	STM32F429ZIYx	Active	8.2333			WLCSP 143 5.6x4.6x0.6 P...	2048 kB...	256 kByt...	114 180 MHz
☆	STM32F437AIH6	STM32F437AI...	Active	10.2559			UFBGA 169 7x7x0.6 P 0.5...	2048 kB...	256 kByt...	130 180 MHz
☆	STM32F437AIH6	STM32F437AI...	Active	10.2559			UFBGA 169 7x7x0.6 P 0.5...	2048 kB...	256 kByt...	130 180 MHz

< Back Next > Finish Cancel

Figura 53. Selección de tarjetas.

Los siguientes pasos son pulsar en “Next”, proporcionarle un nombre al proyecto, como por ejemplo “H2DRONE”, y pulsar “Finish”.

### **3. CONFIGURACIÓN DE PERIFÉRICOS**

Tras la creación del nuevo proyecto, el siguiente paso es configurar los periféricos que se desean emplear. En el archivo .ioc que aparece en la ventana “Project Explorer”, se debe activar y configurar correctamente dos canales DAC, 11 canales ADC, las comunicaciones, y las 7 entradas y 6 salidas digitales.

En primer lugar, se deben activar los dos canales DAC que hay disponibles, habilitando el “buffer” de salida, y configurándolo con DMA circular. Se debe de asegurar que los canales DAC corresponden a los pines PA4 y PA5.

También se deben activar los 11 canales del ADC 2 disponibles, configurándolos con DMA, habilitando el modo SCAN, y eligiendo una resolución de 12 bits, una alineación a derechas y 11 conversiones disparadas por software, como se muestra en la Figura 2:

✓ ADC_Settings		
Clock Prescaler		PCLK2 divided by 4
Resolution		12 bits (15 ADC Clock cycles)
Data Alignment		Right alignment
Scan Conversion Mode		Enabled
Continuous Conversion Mode		Enabled
Discontinuous Conversion Mode		Disabled
DMA Continuous Requests		Disabled
End Of Conversion Selection		EOC flag at the end of single channel conversion
✓ ADC_Regular_ConversionMode		
Number Of Conversion		11
External Trigger Conversion Source		Regular Conversion launched by software
External Trigger Conversion Edge		None

**Figura 54. Configuración ADC.**

Se deben activar seis salidas y seis entradas digitales en modo “push-pull” y sin “pull-up” ni “pull-down”.

## Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

---

La comunicación se debe realizar con la UART 5 y la UART 7, con las siguientes configuraciones:

∨ Basic Parameters	
Baud Rate	115200 Bits/s
Word Length	8 Bits (including Parity)
Parity	None
Stop Bits	1
∨ Advanced Parameters	
Data Direction	Receive and Transmit
Over Sampling	16 Samples

**Figura 55. Configuración UART.**

Una vez finalizada la configuración de periféricos, se debe de guardar el proyecto, y se generará en el archivo “main.c” el código correspondiente a las configuraciones realizadas.

#### 4. CÓDIGO Y MODO RUN

Una vez codificadas todas las configuraciones, se debe pasar el código que se muestra en el Anexo I al nuevo proyecto creado.

En el archivo main.h, que también se debe de pasar, se encuentran las definiciones de constantes(#define), que es donde se van a cambiar los valores de las ganancias de los canales ADC. Estas ganancias permiten la conversión de datos (bits → Amperios, °C, bar...) y dependen de la precisión en el ajuste de los potenciómetros.

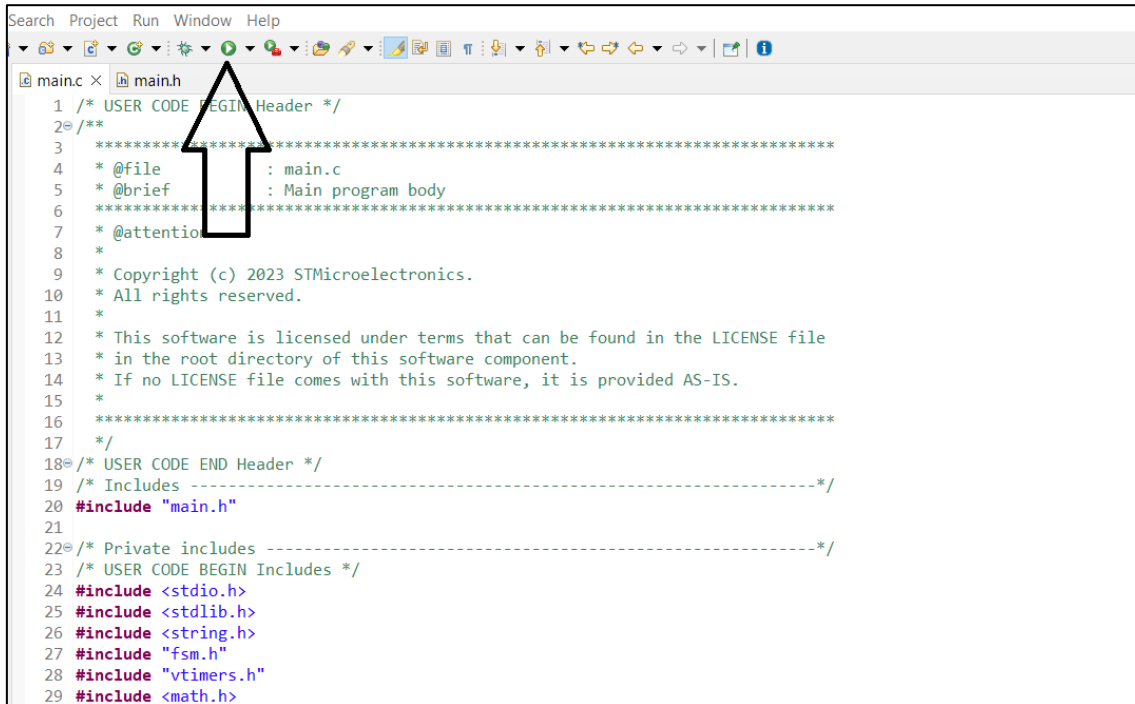
En el main.h también es posible cambiar los valores máximos de seguridad de los parámetros de la pila, el número de celdas de la pila, o los tiempos de ejecución de las máquinas de estados, como se observa en la Figura 3:

```
96 //NUMERO DE CELDAS DE LA PILA DE COMBUSTIBLE
97 #define NUMERO_CELDAS 20
98
99 //NUMERO DE CANALES ADC
100 #define CANALES_ADC 11
101
102 //RESOLUCION ADC
103 #define RESOLUCION 4095.00
104
105 //NUMERO DE SENSORES DE CORRIENTE
106 #define SENSORES_DE_CORRIENTE 2
107
108
109 //TIEMPO DE EJECUCION FSM
110 #define T_FIRE 2500
111 //TIEMPO DE APERTURA DE LA VÁLVULA DE PURGADO C4
112 #define TIEMPO_APERTURA_C4 500
113
114
115 //TENSIONES MÁXIMAS
116 //MÁXIMA TENSIÓN DE CARGA DE LA BATERÍA
117 #define MAX_TENSION_CARGA 16.59
```

Figura 56. main.h del proyecto.

Una vez finalizado el ajuste, el último paso es el modo **RUN**, como se observa en la Figura 5, para “cargar” el programa de control en el microcontrolador, y poder así desconectarlo del ordenador y conectarlo a la placa de interfaz de hardware.

# Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”



```
1 /* USER CODE BEGIN Header */
2 /**
3  * @file          : main.c
4  * @brief        : Main program body
5  * @attention
6  *
7  * Copyright (c) 2023 STMicroelectronics.
8  * All rights reserved.
9  *
10 * This software is licensed under terms that can be found in the LICENSE file
11 * in the root directory of this software component.
12 * If no LICENSE file comes with this software, it is provided AS-IS.
13 *
14 *
15 *
16 *
17 */
18 /* USER CODE END Header */
19 /* Includes -----*/
20 #include "main.h"
21
22 /* Private includes -----*/
23 /* USER CODE BEGIN Includes */
24 #include <stdio.h>
25 #include <stdlib.h>
26 #include <string.h>
27 #include "fsm.h"
28 #include "vtimers.h"
29 #include <math.h>
```

**Figura 57. Modo RUN.**





UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



# **ANEXO IV**

## **Hojas de datos de los componentes**

**Implementación de Máquina de  
Estados Finita en Microcontrolador  
para optimizar el control del sistema  
de energía de un dron híbrido “pila  
de combustible – batería”**





## **ÍNDICE DEL ANEXO IV**

<b>1. HOJA DE DATOS DE LA PLACA DE DESARROLLO STM32F429I-DISC1.....</b>	<b>1</b>
<b>2. HOJA DE DATOS DEL MICROCONTROLADOR STM32F429ZIT6 .....</b>	<b>2</b>
<b>3. HOJA DE DATOS DEL SENSOR DE CORRIENTE ACS712 .....</b>	<b>4</b>
<b>4. HOJA DE DATOS DEL MAX232.....</b>	<b>6</b>
<b>5. HOJA DE DATOS DEL AMPLIFICADOR DE INSTRUMENTACIÓN AD623 .....</b>	<b>8</b>
<b>6. HOJA DE DATOS DEL REGULADOR L7805C.....</b>	<b>13</b>



## 1. Hoja de datos de la placa de desarrollo STM32F429I-DISC1

Features

UM1670

### 1 Features

The STM32F429 Discovery board offers the following features:

- STM32F429ZIT6 microcontroller featuring 2 Mbytes of Flash memory, 256 Kbytes of RAM in an LQFP144 package
- 2.4" QVGA TFT LCD
- USB OTG
- 64-Mbit SDRAM
- L3GD20, ST-MEMS motion sensor 3-axis digital output gyroscope
- Six LEDs:
  - LD1 (red/green) for USB communication
  - LD2 (red) for 3.3 V power-on
  - Two user LEDs: LD3 (green), LD4 (red)
  - Two USB OTG LEDs: LD5 (green) VBUS and LD6 (red) OC (overcurrent)
- Two push-buttons (user and reset)
- USB OTG with Micro-AB connector
- Extension header for LQFP144 I/Os for a quick connection to the prototyping board and an easy probing
- Flexible power-supply options: ST-LINK USB  $V_{BUS}$  or external sources
- On-board ST-LINK/V2 on 32F429IDISCOVERY or ST-LINK/V2-B on STM32F429I-DISC1 with USB re-enumeration capability: mass storage (with ST-LINK/V2-B only), virtual COM port (with ST-LINK/V2-B only) and debug port
- Comprehensive free software including a variety of examples, part of the STM32CubeF4 package or STSW-STM32138 for legacy standard libraries usage
- Arm® Mbed Enabled™ compliant only for ST-LINK/V2-B only

## 2. Hoja de datos del microcontrolador STM32F429ZIT6



life.augmented

### STM32F427xx STM32F429xx

32b Arm® Cortex®-M4 MCU+FPU, 225DMIPS, up to 2MB Flash/256+4KB RAM, USB OTG HS/FS, Ethernet, 17 TIMs, 3 ADCs, 20 com. interfaces, camera & LCD-TFT

Datasheet - production data

#### Features

- Core: Arm® 32-bit Cortex®-M4 CPU with FPU, Adaptive real-time accelerator (ART Accelerator™) allowing 0-wait state execution from Flash memory, frequency up to 180 MHz, MPU, 225 DMIPS/1.25 DMIPS/MHz (Dhystone 2.1), and DSP instructions
  - Memories
    - Up to 2 MB of Flash memory organized into two banks allowing read-while-write
    - Up to 256+4 KB of SRAM including 64-KB of CCM (core coupled memory) data RAM
    - Flexible external memory controller with up to 32-bit data bus: SRAM, PSRAM, SDRAM/LPDDR SDRAM, Compact Flash/NOR/NAND memories
  - LCD parallel interface, 8080/6800 modes
  - LCD-TFT controller with fully programmable resolution (total width up to 4096 pixels, total height up to 2048 lines and pixel clock up to 83 MHz)
  - Chrom-ART Accelerator™ for enhanced graphic content creation (DMA2D)
  - Clock, reset and supply management
    - 1.7 V to 3.6 V application supply and I/Os
    - POR, PDR, PVD and BOR
    - 4-to-26 MHz crystal oscillator
    - Internal 16 MHz factory-trimmed RC (1% accuracy)
    - 32 kHz oscillator for RTC with calibration
    - Internal 32 kHz RC with calibration
  - Low power
    - Sleep, Stop and Standby modes
    - V<sub>BAT</sub> supply for RTC, 20×32 bit backup registers + optional 4 KB backup SRAM
  - 3×12-bit, 2.4 MSPS ADC: up to 24 channels and 7.2 MSPS in triple interleaved mode
  - 2×12-bit D/A converters
  - General-purpose DMA: 16-stream DMA controller with FIFOs and burst support
  - Up to 17 timers: up to twelve 16-bit and two 32-bit timers up to 180 MHz, each with up to 4 IC/OC/PWM or pulse counter and quadrature (incremental) encoder input
- LQFP100 (14 × 14 mm) UFBGA176 (10 × 10 mm) WLCSP143  
LQFP144 (20 × 20 mm) UFBGA169 (7 × 7 mm)  
LQFP176 (24 × 24 mm) TFPGA216 (13 × 13 mm)  
LQFP208 (28 × 28 mm)
- Debug mode
    - SWD & JTAG interfaces
    - Cortex-M4 Trace Macrocell™
  - Up to 168 I/O ports with interrupt capability
    - Up to 164 fast I/Os up to 90 MHz
    - Up to 166 5 V-tolerant I/Os
  - Up to 21 communication interfaces
    - Up to 3 × I<sup>2</sup>C interfaces (SMBus/PMBus)
    - Up to 4 USARTs/4 UARTs (11.25 Mbit/s, ISO7816 interface, LIN, IrDA, modem control)
    - Up to 6 SPIs (45 Mbits/s), 2 with muxed full-duplex I<sup>2</sup>S for audio class accuracy via internal audio PLL or external clock
    - 1 × SAI (serial audio interface)
    - 2 × CAN (2.0B Active) and SDIO interface
  - Advanced connectivity
    - USB 2.0 full-speed device/host/OTG controller with on-chip PHY
    - USB 2.0 high-speed/full-speed device/host/OTG controller with dedicated DMA, on-chip full-speed PHY and ULPI
    - 10/100 Ethernet MAC with dedicated DMA: supports IEEE 1588v2 hardware, MII/RMII
  - 8- to 14-bit parallel camera interface up to 54 Mbytes/s
  - True random number generator
  - CRC calculation unit
  - RTC: subsecond accuracy, hardware calendar
  - 96-bit unique ID

### 3.37 General-purpose input/outputs (GPIOs)

Each of the GPIO pins can be configured by software as output (push-pull or open-drain, with or without pull-up or pull-down), as input (floating, with or without pull-up or pull-down) or as peripheral alternate function. Most of the GPIO pins are shared with digital or analog alternate functions. All GPIOs are high-current-capable and have speed selection to better manage internal noise, power consumption and electromagnetic emission.

The I/O configuration can be locked if needed by following a specific sequence in order to avoid spurious writing to the I/Os registers.

Fast I/O handling allowing maximum I/O toggling up to 90 MHz.

### 3.38 Analog-to-digital converters (ADCs)

Three 12-bit analog-to-digital converters are embedded and each ADC shares up to 16 external channels, performing conversions in the single-shot or scan mode. In scan mode, automatic conversion is performed on a selected group of analog inputs.

Additional logic functions embedded in the ADC interface allow:

- Simultaneous sample and hold
- Interleaved sample and hold

The ADC can be served by the DMA controller. An analog watchdog feature allows very precise monitoring of the converted voltage of one, some or all selected channels. An interrupt is generated when the converted voltage is outside the programmed thresholds.

To synchronize A/D conversion and timers, the ADCs could be triggered by any of TIM1, TIM2, TIM3, TIM4, TIM5, or TIM8 timer.

### 3.40 Digital-to-analog converter (DAC)

The two 12-bit buffered DAC channels can be used to convert two digital signals into two analog voltage signal outputs.

This dual digital Interface supports the following features:

- two DAC converters: one for each output channel
- 8-bit or 10-bit monotonic output
- left or right data alignment in 12-bit mode
- synchronized update capability
- noise-wave generation
- triangular-wave generation
- dual DAC channel independent or simultaneous conversions
- DMA capability for each channel
- external triggers for conversion
- input voltage reference  $V_{REF+}$

Eight DAC trigger inputs are used in the device. The DAC channels are triggered through the timer update outputs that are also connected to different DMA streams.

### 3. Hoja de datos del sensor de corriente ACS712

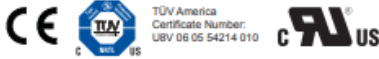


## ACS712

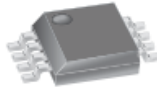
*Fully Integrated, Hall Effect-Based Linear Current Sensor IC  
with 2.1 kVRMS Isolation and a Low-Resistance Current Conductor*

#### Features and Benefits

- Low-noise analog signal path
- Device bandwidth is set via the new FILTER pin
- 5  $\mu$ s output rise time in response to step input current
- 80 kHz bandwidth
- Total output error 1.5% at  $T_A = 25^\circ\text{C}$
- Small footprint, low-profile SOIC8 package
- 1.2 m $\Omega$  internal conductor resistance
- 2.1 kVRMS minimum isolation voltage from pins 1-4 to pins 5-8
- 5.0 V, single supply operation
- 66 to 185 mV/A output sensitivity
- Output voltage proportional to AC or DC currents
- Factory-trimmed for accuracy
- Extremely stable output offset voltage
- Nearly zero magnetic hysteresis
- Ratiometric output from supply voltage



Package: 8 Lead SOIC (suffix LC)



Approximate Scale 1:1

#### Description

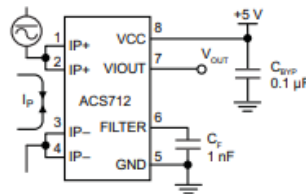
The Allegro™ ACS712 provides economical and precise solutions for AC or DC current sensing in industrial, commercial, and communications systems. The device package allows for easy implementation by the customer. Typical applications include motor control, load detection and management, switch-mode power supplies, and overcurrent fault protection. The device is not intended for automotive applications.

The device consists of a precise, low-offset, linear Hall circuit with a copper conduction path located near the surface of the die. Applied current flowing through this copper conduction path generates a magnetic field which the Hall IC converts into a proportional voltage. Device accuracy is optimized through the close proximity of the magnetic signal to the Hall transducer. A precise, proportional voltage is provided by the low-offset, chopper-stabilized BiCMOS Hall IC, which is programmed for accuracy after packaging.

The output of the device has a positive slope ( $>V_{\text{IOUT}(Q)}$ ) when an increasing current flows through the primary copper conduction path (from pins 1 and 2, to pins 3 and 4), which is the path used for current sampling. The internal resistance of this conductive path is 1.2 m $\Omega$  typical, providing low power loss. The thickness of the copper conductor allows survival of

*Continued on the next page...*

#### Typical Application



Application 1. The ACS712 outputs an analog signal,  $V_{\text{OUT}}$ , that varies linearly with the uni- or bi-directional AC or DC primary sampled current,  $I_P$ , within the range specified.  $C_F$  is recommended for noise management, with values that depend on the application.

Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

## ACS712

Fully Integrated, Hall Effect-Based Linear Current Sensor IC  
with 2.1 kVRMS Isolation and a Low-Resistance Current Conductor

### COMMON OPERATING CHARACTERISTICS<sup>1</sup> over full range of $T_A$ , $C_F = 1$ nF, and $V_{CC} = 5$ V, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
<b>ELECTRICAL CHARACTERISTICS</b>						
Supply Voltage	$V_{CC}$		4.5	5.0	5.5	V
Supply Current	$I_{CC}$	$V_{CC} = 5.0$ V, output open	–	10	13	mA
Output Capacitance Load	$C_{LOAD}$	V <sub>IOUT</sub> to GND	–	–	10	nF
Output Resistive Load	$R_{LOAD}$	V <sub>IOUT</sub> to GND	4.7	–	–	kΩ
Primary Conductor Resistance	$R_{PRIMARY}$	$T_A = 25^\circ\text{C}$	–	1.2	–	mΩ
Rise Time	$t_r$	$I_P = I_P(\text{max})$ , $T_A = 25^\circ\text{C}$ , $C_{OUT} = \text{open}$	–	3.5	–	μs
Frequency Bandwidth	$f$	–3 dB, $T_A = 25^\circ\text{C}$ ; $I_P$ is 10 A peak-to-peak	–	80	–	kHz
Nonlinearity	$E_{LIN}$	Over full range of $I_P$	–	1.5	–	%
Symmetry	$E_{SYM}$	Over full range of $I_P$	98	100	102	%
Zero Current Output Voltage	$V_{IOUT(O)}$	Bidirectional; $I_P = 0$ A, $T_A = 25^\circ\text{C}$	–	$V_{CC} \times 0.5$	–	V
Power-On Time	$t_{PO}$	Output reaches 90% of steady-state level, $T_J = 25^\circ\text{C}$ , 20 A present on leadframe	–	35	–	μs
Magnetic Coupling <sup>2</sup>			–	12	–	G/A
Internal Filter Resistance <sup>3</sup>	$R_{F(INT)}$			1.7		kΩ

<sup>1</sup>Device may be operated at higher primary current levels,  $I_P$ , and ambient,  $T_A$ , and internal leadframe temperatures,  $T_A$ , provided that the Maximum Junction Temperature,  $T_J(\text{max})$ , is not exceeded.

<sup>2</sup>1G = 0.1 mT.

<sup>3</sup> $R_{F(INT)}$  forms an RC circuit via the FILTER pin.

### COMMON THERMAL CHARACTERISTICS<sup>1</sup>

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Operating Internal Leadframe Temperature	$T_A$	E range	–40	–	85	$^\circ\text{C}$
Junction-to-Lead Thermal Resistance <sup>2</sup>	$R_{\theta JL}$	Mounted on the Allegro ASEK 712 evaluation board			5	$^\circ\text{C/W}$
Junction-to-Ambient Thermal Resistance	$R_{\theta JA}$	Mounted on the Allegro 85-0322 evaluation board, includes the power consumed by the board			23	$^\circ\text{C/W}$

<sup>1</sup>Additional thermal information is available on the Allegro website.

<sup>2</sup>The Allegro evaluation board has 1500 mm<sup>2</sup> of 2 oz. copper on each side, connected to pins 1 and 2, and to pins 3 and 4, with thermal vias connecting the layers. Performance values include the power consumed by the PCB. Further details on the board are available from the Frequently Asked Questions document on our website. Further information about board design and thermal performance also can be found in the Applications Information section of this datasheet.

### x30A PERFORMANCE CHARACTERISTICS<sup>1</sup> $T_A = -40^\circ\text{C}$ to $85^\circ\text{C}$ , $C_F = 1$ nF, and $V_{CC} = 5$ V, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Optimized Accuracy Range	$I_P$		–30	–	30	A
Sensitivity	Sens	Over full range of $I_P$ , $T_A = 25^\circ\text{C}$	63	66	69	mV/A
Noise	$V_{NOISE(PP)}$	Peak-to-peak, $T_A = 25^\circ\text{C}$ , 66 mV/A programmed Sensitivity, $C_F = 47$ nF, $C_{OUT} = \text{open}$ , 2 kHz bandwidth	–	7	–	mV
Zero Current Output Slope	$\Delta V_{OUT(O)}$	$T_A = -40^\circ\text{C}$ to $25^\circ\text{C}$	–	–0.35	–	mV/ $^\circ\text{C}$
		$T_A = 25^\circ\text{C}$ to $150^\circ\text{C}$	–	–0.08	–	mV/ $^\circ\text{C}$
Sensitivity Slope	$\Delta \text{Sens}$	$T_A = -40^\circ\text{C}$ to $25^\circ\text{C}$	–	0.007	–	mV/A/ $^\circ\text{C}$
		$T_A = 25^\circ\text{C}$ to $150^\circ\text{C}$	–	–0.002	–	mV/A/ $^\circ\text{C}$
Total Output Error <sup>2</sup>	$E_{TOT}$	$I_P = \pm 30$ A, $T_A = 25^\circ\text{C}$	–	$\pm 1.5$	–	%

<sup>1</sup>Device may be operated at higher primary current levels,  $I_P$ , and ambient temperatures,  $T_A$ , provided that the Maximum Junction Temperature,  $T_J(\text{max})$ , is not exceeded.

<sup>2</sup>Percentage of  $I_P$ , with  $I_P = 30$  A. Output filtered.



## 4. Hoja de datos del MAX232

### MAX13223E

### ±70V Fault-Protected, 3.0V to 5.5V, 2Tx/2Rx RS-232 Transceiver

#### General Description

The MAX13223E is a +3.0V to +5.5V-powered EIA/TIA-232 and V.28 communications interface with fault protection on the RS-232 line interface. This allows shorts of the transmitter outputs and receiver inputs to voltages in the ±70V range without adversely affecting the MAX13223E.

The MAX13223E achieves 1µA supply current using Maxim's AutoShutdown™ feature. The MAX13223E automatically enters a low-power shutdown mode when the RS-232 cable is disconnected or the receivers are inactive. The device turns on again when a valid transition at any receiver input is sensed. A proprietary, high-efficiency, dual charge-pump power supply and a low-dropout transmitter combine to deliver true RS-232 performance from a single +3.0V to +5.5V supply.

The MAX13223E has two receivers and two drivers and is guaranteed to run at data rates of 250kbps for one transmitter switching while maintaining RS-232 output levels.

The MAX13223E operates from input voltages ranging from +3.0V to +5.5V and is available in a 20-pin, 6.5mm x 4.4mm, TSSOP package. The MAX13223E is specified over the -40°C to +85°C temperature range.

#### Applications

- Telematics Equipment
- Base Stations
- Utility Meters
- Industrial Equipment
- Telecomm Equipment
- POS Terminal Equipment

*AutoShutdown is a trademark of Maxim Integrated Products, Inc.*

#### Benefits and Features

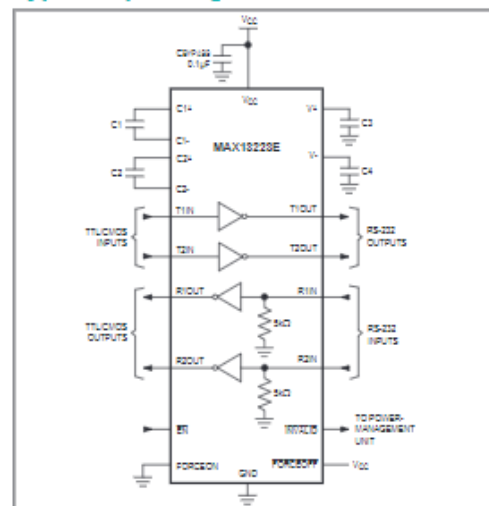
- Robust Protection to Withstand Load Dumps, Miswiring, and Connector Failures
  - ±70V Fault Protection
  - Overvoltage Current Limiting
  - Current Protection at Transmitter Outputs
  - Thermal Shutdown Protection
- Flexible Power Requirements Simplify Design
  - +3.0V to +5.5V Single Supply Voltage
  - True RS-232 Performance (Up to 250kbps) Even from +3.0V Supply Voltage
- AutoShutdown Simplifies Power Savings
  - Automatically Enters Shutdown Mode if RS-232 Cable is Disconnected or Receivers are Inactive
  - Automatically Activates from Shutdown When Any Valid Transition is Sensed
  - 1µA Typical Shutdown Current

#### Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX13223EEUP+	-40°C to +85°C	20 TSSOP

+Denotes a lead(Pb)-free/RoHS-compliant package.

#### Typical Operating Circuit



# Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

MAX13223E

±70V Fault-Protected, 3.0V to 5.5V,  
2Tx/2Rx RS-232 Transceiver

## Absolute Maximum Ratings

(Voltages referenced to GND.)

V <sub>CC</sub>	-0.3V to +7.0V
V+	-0.3V to +7.0V
V-	+0.3V to -7.0V
V+ to V-	+13V
Input Voltages	
T1IN, T2IN, EN, FORCEON, FORCEOFF	-0.3V to +6.0V
R1IN, R2IN	±70V
Output Voltages	
T1OUT, T2OUT	±70V
R1OUT, R2OUT, INVALID	-0.3V to (V <sub>CC</sub> + 0.3V)

Short-Circuit Duration

T1OUT, T2OUT	Continuous
Continuous Power Dissipation (T <sub>A</sub> = +70°C)	
TSSOP (derate 13.6mW/°C above +70°C)	1084mW
Operating Temperature Range	-40°C to +85°C
Junction Temperature	+150°C
Storage Temperature Range	-65°C to +150°C
Lead Temperature (soldering, 10s)	+300°C
Soldering Temperature (reflow)	+260°C

Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## Package Thermal Characteristics (Note 1)

TSSOP

Junction-to-Ambient Thermal Resistance (θ <sub>JA</sub> )	73.8°C/W
Junction-to-Case Thermal Resistance (θ <sub>JC</sub> )	20°C/W

**Note 1:** Package thermal resistances were obtained using the method described in JEDEC specification JESD51-7, using a four-layer board. For detailed information on package thermal considerations, refer to [www.maximintegrated.com/thermal-tutorial](http://www.maximintegrated.com/thermal-tutorial).

## Electrical Characteristics

(V<sub>CC</sub> = +3.0V to +5.5V, T<sub>A</sub> = -40°C to +85°C, unless otherwise noted. Typical values are at V<sub>CC</sub> = +3.3V, T<sub>A</sub> = +25°C, unless otherwise noted. For V<sub>CC</sub> = +3.0V to +3.6V, C1 = C2 = C3 = 0.1µF, C4 = 1µF. For V<sub>CC</sub> = +4.5V to +5.5V, C1 = 47nF, C2 = C3 = 330nF, C4 = 1µF.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V <sub>CC</sub>		3.0		5.5	V
Supply Current		AutoShutdown disabled (FORCEON = FORCEOFF = V <sub>CC</sub> ), no load		8	15	mA
Supply Current AutoShutdown	I <sub>ASD</sub>	FORCEON = GND, FORCEOFF = V <sub>CC</sub> , R1IN and R2IN idle, T1IN and T2IN idle		1.0	10	µA
Supply Current Shutdown	I <sub>SD</sub>	FORCEOFF = GND		1.0	10	µA
<b>LOGIC INPUTS</b>						
Input-Logic Low	V <sub>T,IN,LO</sub>	T <sub>IN</sub> , EN, FORCEON, FORCEOFF, V <sub>CC</sub> = +3.3V to +3.6V, +5.0V to +5.5V			0.8	V
Input-Logic High	V <sub>T,IN,HI</sub>	T <sub>IN</sub> , FORCEON, FORCEOFF, EN		2.0		V
		V <sub>CC</sub> = +3.3V		2.4		
		V <sub>CC</sub> = +5.0V				
Transmitter Input Hysteresis	V <sub>TX,INHYS</sub>			0.5		V
Input Leakage Current	I <sub>IN,LKG</sub>	T <sub>IN</sub> , EN, FORCEON, FORCEOFF		±0.01	±1	µA
<b>RECEIVER OUTPUTS</b>						
Output Leakage Current	I <sub>RX,OUT,LKG</sub>	EN = V <sub>CC</sub>		±0.05	±10	µA
Output-Voltage Low	V <sub>RX,OUT,LO</sub>	I <sub>OUT</sub> = 1.6mA			0.4	V
Output-Voltage High	V <sub>RX,OUT,HI</sub>	I <sub>OUT</sub> = -1.0mA	V <sub>CC</sub> - 0.6		V <sub>CC</sub> - 0.2	V

## 5. Hoja de datos del Amplificador de Instrumentación AD623



### Single and Dual-Supply, Rail-to-Rail, Low Cost Instrumentation Amplifier

Data Sheet

AD623

#### FEATURES

- Easy to use
- Rail-to-rail output swing
- Input voltage range extends 150 mV below ground (single supply)
- Low power, 550  $\mu$ A maximum quiescent current
- Gain set with one external resistor
  - Gain range: 1 to 1000
- High accuracy dc performance
  - 0.10% gain error ( $G = 1$ )
  - 0.35% gain error ( $G > 1$ )
- Noise: 35 nV/ $\sqrt{\text{Hz}}$  RTI noise at 1 kHz
- Optimal dynamic specifications
  - 800 kHz bandwidth ( $G = 1$ )
  - 20  $\mu$ s settling time to 0.01% ( $G = 10$ )

#### APPLICATIONS

- Low power medical instrumentation
- Transducer interfaces
- Thermocouple amplifiers
- Industrial process controls
- Difference amplifiers
- Low power data acquisition

#### GENERAL DESCRIPTION

The AD623 is an integrated, single- or dual-supply instrumentation amplifier that delivers rail-to-rail output swing using supply voltages from 2.7 V to 12 V. The AD623 offers user flexibility by allowing single gain set resistor programming and by conforming to the 8-lead industry standard pinout configuration. With no external resistor, the AD623 is configured for unity gain ( $G = 1$ ), and with an external resistor, the AD623 can be programmed for gains of up to 1000.

The accuracy of the AD623 is the result of increasing ac common-mode rejection ratio (CMRR) coincident with increasing gain. Line noise harmonics are rejected due to constant CMRR up to 200 Hz. The AD623 has a wide input common-mode range and amplifies signals with common-mode voltages as low as 150 mV below ground. The AD623 maintains optimal performance with dual and single polarity power supplies.

Table 1. Low Power Upgrades for the AD623

Part No.	Total Supply Voltage, $V_S$ (V dc)	Typical Quiescent Current, $I_Q$ ( $\mu$ A)
AD8235	5.5	30
AD8236	5.5	33
AD8237	5.5	33
AD8226	36	350
AD8227	36	325
AD8420	36	85
AD8422	36	300
AD8426	36	325 (per channel)

#### FUNCTIONAL BLOCK DIAGRAM

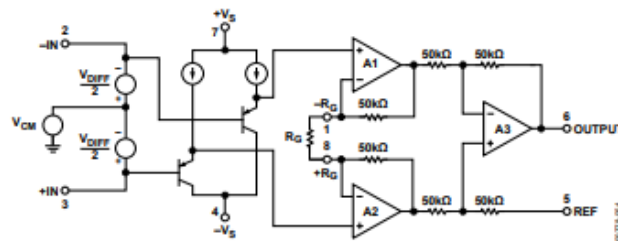


Figure 1.

Rev. G [Document Feedback](#)  
 Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.  
 Tel: 781.329.4700 ©2020 Analog Devices, Inc. All rights reserved.  
[Technical Support](#) [www.analog.com](http://www.analog.com)

Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

AD623

Data Sheet

**SPECIFICATIONS**

**SINGLE SUPPLY**

Typical at 25°C, single supply, +V<sub>S</sub> = 5 V, -V<sub>S</sub> = 0 V, and load resistance (R<sub>L</sub>) = 10 kΩ, unless otherwise noted.

Table 2.

Parameter	Test Conditions/ Comments	AD623ANZ, AD623ARZ			AD623ARM			AD623BNZ, AD623BRZ			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
<b>GAIN</b>											
Gain Range	G = 1 + (100 k/external resistor (R <sub>C</sub> ))	1		1000	1		1000	1		1000	
Gain Error <sup>1</sup>	G1 output voltage (V <sub>out</sub> ) = 0.15 V to 3.5 V G > 1 V <sub>out</sub> = 0.15 V to 4.5 V										
G = 1			0.03	0.10		0.03	0.10		0.03	0.05	%
G = 10			0.10	0.35		0.10	0.35		0.10	0.35	%
G = 100			0.10	0.35		0.10	0.35		0.10	0.35	%
G = 1000			0.10	0.35		0.10	0.35		0.10	0.35	%
Nonlinearity	G1 V <sub>out</sub> = 0.15 V to 3.5 V G > 1 V <sub>out</sub> = 0.15 V to 4.5 V										
G = 1 to 1000			50			50			50		ppm
Gain vs. Temperature											
G = 1			5	10		5	10		5	10	ppm/°C
G > 1 <sup>1</sup>			50			50			50		ppm/°C
<b>VOLTAGE OFFSET</b>											
Total referred to input (RTI) error = V <sub>os1</sub> + V <sub>os2</sub> /G											
Input Offset, V <sub>os1</sub>			25	200		200	500		25	100	μV
Over Temperature				350			650			160	μV
Average Temperature Coefficient (Tempco)			0.1	2		0.1	2		0.1	1	μV/°C
Output Offset, V <sub>os2</sub>			200	1000		500	2000		200	500	μV
Over Temperature				1500			2600			1100	μV
Average Tempco			2.5	10		2.5	10		2.5	10	μV/°C
Offset Referred to the Input vs. Supply (PSR)											
G = 1		80	100		80	100		80	100		dB
G = 10		100	120		100	120		100	120		dB
G = 100		100	130		100	130		100	130		dB
G = 1000		100	130		100	130		100	130		dB
<b>INPUT CURRENT</b>											
Input Bias Current			17	25		17	25		17	25	nA
Over Temperature				27.5			27.5			27.5	nA
Average Tempco			25			25			25		pA/°C
Input Offset Current			0.25	2		0.25	2		0.25	2	nA
Over Temperature				2.5			2.5			2.5	nA
Average Tempco			5			5			5		pA/°C

Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

Data Sheet											AD623
Parameter	Test Conditions/ Comments	AD623ANZ, AD623ARZ			AD623ARM			AD623BNZ, AD623BRZ			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
<b>INPUT</b>											
Input Impedance			2  2		2  2		2  2		2  2		GΩ  pF
Differential			2  2		2  2		2  2		2  2		GΩ  pF
Common-Mode			2  2		2  2		2  2		2  2		GΩ  pF
Input Voltage Range <sup>2</sup>	V <sub>S</sub> = 3 V to 12 V	(-V <sub>S</sub> ) – 0.15		(+V <sub>S</sub> ) – 1.5	(-V <sub>S</sub> ) – 0.15		(+V <sub>S</sub> ) – 1.5	(-V <sub>S</sub> ) – 0.15		(+V <sub>S</sub> ) – 1.5	V
Common-Mode Rejection at 60 Hz with 1 kΩ Source Imbalance											
G = 1	Common-mode voltage (V <sub>CM</sub> ) = 0 V to 3 V	70	80		70	80		77	86		dB
G = 10	V <sub>CM</sub> = 0 V to 3 V	90	100		90	100		94	100		dB
G = 100	V <sub>CM</sub> = 0 V to 3 V	105	110		105	110		105	110		dB
G = 1000	V <sub>CM</sub> = 0 V to 3 V	105	110		105	110		105	110		dB
<b>OUTPUT</b>											
Output Swing	R <sub>L</sub> = 10 kΩ	0.2		(+V <sub>S</sub> ) – 0.5	0.2		(+V <sub>S</sub> ) – 0.5	0.2		(+V <sub>S</sub> ) – 0.5	V
	R <sub>L</sub> = 100 kΩ	0.05		(+V <sub>S</sub> ) – 0.15	0.05		(+V <sub>S</sub> ) – 0.15	0.05		(+V <sub>S</sub> ) – 0.15	V
<b>DYNAMIC RESPONSE</b>											
Small Signal –3 dB Bandwidth											
G = 1			800		800		800		800		kHz
G = 10			100		100		100		100		kHz
G = 100			10		10		10		10		kHz
G = 1000			2		2		2		2		kHz
Slew Rate			0.3		0.3		0.3		0.3		V/μs
Settling Time to 0.01%	V <sub>S</sub> = 5 V										
G = 1	Step size = 3.5 V		30		30		30		30		μs
G = 10	Step size = 4 V, V <sub>CM</sub> = 1.8 V		20		20		20		20		μs

<sup>1</sup> Does not include effects of external resistor, R<sub>L</sub>.

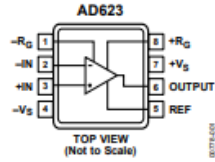
<sup>2</sup> One input grounded. G = 1.

Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

AD623

Data Sheet

**PIN CONFIGURATION AND FUNCTION DESCRIPTIONS**



**Table 6. Pin Function Descriptions**

Pin No.	Mnemonic	Description
1	-R <sub>G</sub>	Inverting Terminal of External Gain Setting Resistor, R <sub>G</sub> .
2	-IN	Inverting In-Amp Input.
3	+IN	Noninverting In-Amp Input.
4	-V <sub>S</sub>	Negative Supply Terminal.
5	REF	In-Amp Output Reference Input. The voltage input establishes the common-mode voltage of the output.
6	OUTPUT	In-Amp Output.
7	+V <sub>S</sub>	Positive Supply Terminal.
8	+R <sub>G</sub>	Noninverting Terminal of External Gain Setting Resistor, R <sub>G</sub> .

**APPLICATIONS INFORMATION**  
**BASIC CONNECTION**

Figure 71 and Figure 72 show the basic connection circuits for the AD623. The +V<sub>S</sub> and -V<sub>S</sub> terminals are connected to the power supply. The supply can be either bipolar (V<sub>S</sub> = ±2.5 V to ±6 V) or single supply (-V<sub>S</sub> = 0 V, +V<sub>S</sub> = 2.7 V to 12 V). Capacitively decouple power supplies close to the power pins of the device. For optimal results, use surface-mount 0.1 μF ceramic chip capacitors and 10 μF electrolytic tantalum capacitors.

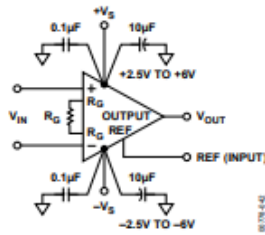


Figure 71. Dual-Supply Basic Connection

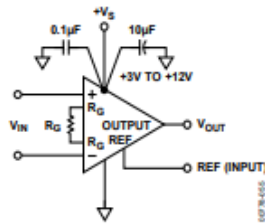


Figure 72. Single-Supply Basic Connection

The input voltage, which can be either single-ended (tie either -IN or +IN to ground) or differential, is amplified by the programmed gain. The output signal appears as the voltage difference between the OUTPUT pin and the externally applied voltage on the REF input. For a ground referenced output, ground REF.

**GAIN SELECTION**

The gain of the AD623 is programmed by the R<sub>G</sub> resistor, or more precisely, by whatever impedance appears between Pin 1 and Pin 8. The AD623 offers accurate gains using 0.1% to 1% tolerance resistors. Table 7 shows the required values of R<sub>G</sub> for the various gains. Note that for G = 1, the R<sub>G</sub> terminals are unconnected (R<sub>G</sub> = ∞). For any arbitrary gain, R<sub>G</sub> can be calculated by

$$R_G = 100 \text{ k}\Omega / (G - 1)$$

Table 7. Required Values of Gain Resistors

Desired Gain	1% Standard Table Value of R <sub>G</sub>	Calculated Gain Using 1% Resistors
2	100 kΩ	2
5	24.9 kΩ	5.02
10	11 kΩ	10.09
20	5.23 kΩ	20.12
33	3.09 kΩ	33.36
40	2.55 kΩ	40.21
50	2.05 kΩ	49.78
65	1.58 kΩ	64.29
100	1.02 kΩ	99.04
200	499 Ω	201.4
500	200 Ω	501
1000	100 Ω	1001

**REFERENCE TERMINAL**

The reference terminal potential defines the zero output voltage and is especially useful when the load does not share a precise ground with the rest of the system. The reference terminal provides a direct means of injecting a precise offset to the output. The reference terminal is also useful when bipolar signals are being amplified because the terminal can provide a virtual ground voltage. The voltage on the reference terminal can vary from -V<sub>S</sub> to +V<sub>S</sub>.

**INPUT AND OUTPUT OFFSET VOLTAGE ERROR**

The offset voltage (V<sub>OS</sub>) of the AD623 is attributed to two sources: those originating in the two input stages where the instrumentation amplifier gain is established, and those originating in the subtractor output stage. The output error is divided by the programmed gain when referred to the input. In practice, the input errors dominate at high gain settings, whereas the output error prevails when the gain is set at or near unity.

Calculate the V<sub>OS</sub> error for any given gain as follows:

$$\begin{aligned} \text{Total Error Referred to Input (RTI)} \\ = \text{Input Error} + (\text{Output Error}/G) \end{aligned}$$

$$\begin{aligned} \text{Total Error Referred to Output (RTO)} \\ = (\text{Input Error} \times G) + \text{Output Error} \end{aligned}$$

The RTI offset errors and noise voltages for different gains are listed in Table 8.

## 6. Hoja de datos del regulador L7805C



**L78xx  
L78xxC**

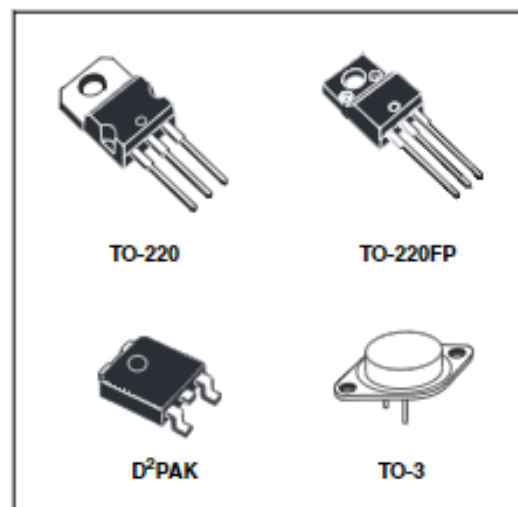
### Positive voltage regulators

#### Features

- Output current to 1.5 A
- Output voltages of 5; 5.2; 6; 8; 8.5; 9; 10; 12; 15; 18; 20; 24 V
- Thermal overload protection
- Short circuit protection
- Output transition SOA protection

#### Description

The L78xx series of three-terminal positive regulators is available in TO-220, TO-220FP, TO-3 and D<sup>2</sup>PAK packages and several fixed output voltages, making it useful in a wide range of applications. These regulators can provide local on-card regulation, eliminating the distribution problems associated with single point regulation. Each type employs internal current limiting, thermal shut-down and safe area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1 A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltage and currents.



**Table 1. Device summary**

Order codes	
L7805	L7810C
L7805C	L7812C
L7852C	L7815C
L7806C	L7818C
L7808C	L7820C
L7885C	L7824C
L7809C	



### 3 Maximum ratings

Table 2. Absolute maximum ratings

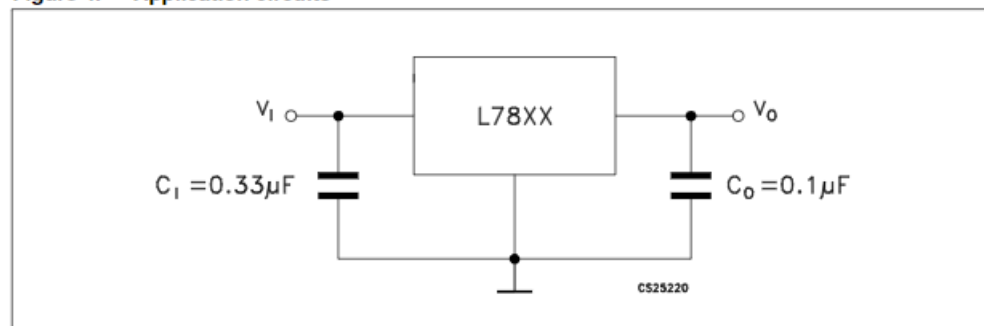
Symbol	Parameter	Value	Unit	
$V_I$	DC Input voltage	for $V_O=5$ to 18 V	35	V
		for $V_O=20, 24$ V	40	
$I_O$	Output current	Internally Limited		
$P_D$	Power dissipation	Internally Limited		
$T_{STG}$	Storage temperature range	-65 to 150	°C	
$T_{OP}$	Operating junction temperature range	for L7800	-55 to 150	°C
		for L7800C	0 to 150	

Note: Absolute Maximum Ratings are those values beyond which damage to the device may occur. Functional operation under these condition is not implied

Table 3. Thermal data

Symbol	Parameter	D <sup>2</sup> PAK	TO-220	TO-220FP	TO-3	Unit
$R_{thJC}$	Thermal resistance junction-case	3	5	5	4	°C/W
$R_{thJA}$	Thermal resistance junction-ambient	62.5	50	60	35	°C/W

Figure 4. Application circuits



Implementación de Máquina de Estados Finita en Microcontrolador para optimizar el control del sistema de energía de un dron híbrido “pila de combustible – batería”

L78xx - L78xxC

Electrical characteristics

**Table 12. Electrical characteristics of L7805C** (refer to the test circuits,  $T_J = 0$  to  $150^\circ\text{C}$ ,  $V_I = 10$  V,  $I_O = 500$  mA,  $C_I = 0.33$   $\mu\text{F}$ ,  $C_O = 0.1$   $\mu\text{F}$  unless otherwise specified)

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
$V_O$	Output voltage	$T_J = 25^\circ\text{C}$	4.8	5	5.2	V
$V_O$	Output voltage	$I_O = 5$ mA to 1 A, $P_O \leq 15$ W $V_I = 7$ to 20 V	4.75	5	5.25	V
$\Delta V_O^{(1)}$	Line regulation	$V_I = 7$ to 25 V, $T_J = 25^\circ\text{C}$		3	100	mV
		$V_I = 8$ to 12 V, $T_J = 25^\circ\text{C}$		1	50	
$\Delta V_O^{(1)}$	Load regulation	$I_O = 5$ mA to 1.5 A, $T_J = 25^\circ\text{C}$			100	mV
		$I_O = 250$ to 750 mA, $T_J = 25^\circ\text{C}$			50	
$I_d$	Quiescent current	$T_J = 25^\circ\text{C}$			8	mA
$\Delta I_d$	Quiescent current change	$I_O = 5$ mA to 1 A			0.5	mA
		$V_I = 7$ to 25 V			0.8	
$\Delta V_O/\Delta T$	Output voltage drift	$I_O = 5$ mA		-1.1		mV/ $^\circ\text{C}$
eN	Output noise voltage	B = 10 Hz to 100 KHz, $T_J = 25^\circ\text{C}$		40		$\mu\text{V}/V_O$
SVR	Supply voltage rejection	$V_I = 8$ to 18 V, $f = 120$ Hz	62			dB
$V_d$	Dropout voltage	$I_O = 1$ A, $T_J = 25^\circ\text{C}$		2		V
$R_O$	Output resistance	$f = 1$ KHz		17		m $\Omega$
$I_{sc}$	Short circuit current	$V_I = 35$ V, $T_J = 25^\circ\text{C}$		0.75		A
$I_{scp}$	Short circuit peak current	$T_J = 25^\circ\text{C}$		2.2		A

1. Load and line regulation are specified at constant junction temperature. Changes in  $V_O$  due to heating effects must be taken into account separately. Pulse testing with low duty cycle is used.