



UNIVERSIDAD
DE MÁLAGA



TRABAJO FIN DE GRADO

DESARROLLO DE UN SISTEMA DE GESTIÓN EFICIENTE DE BATERÍAS PARA UNA MOTOCICLETA ELÉCTRICA DE COMPETICIÓN

GRADO EN INGENIERÍA ELECTRÓNICA, ROBÓTICA Y MECATRÓNICA.

ESCUELA DE INGENIERÍAS INDUSTRIALES

UNIVERSIDAD DE MÁLAGA

Autor: Daniel Vega Moreno

Tutor: Dr. D. Javier Pérez Fernández

Área de conocimiento: Ingeniería electrónica

Cotutor: D. Juan María Velasco García

Área de conocimiento: Ingeniería electrónica

Málaga, junio de 2025

DESARROLLO DE UN SISTEMA DE GESTIÓN EFICIENTE DE BATERÍAS PARA UNA MOTOCICLETA ELÉCTRICA DE COMPETICIÓN

Resumen: En este proyecto se desarrolla un sistema de gestión eficiente de baterías para una motocicleta eléctrica del equipo UMA Racing Team, en el marco de la competición MotoStudent. Se parte de una base electrónica ya implementada en ediciones anteriores y de un código funcional empleado únicamente como sistema de monitorización. A partir de esta base, se reestructura el código hacia una arquitectura orientada a objetos, ampliando sus funcionalidades hasta convertirla en un *battery management system* (BMS) completo con capacidades de monitorización, balanceo, control de carga y protección.

Para contextualizar el trabajo, se realiza un estudio previo sobre baterías de ion-litio y BMS, incluyendo curvas de carga y algoritmos de balanceo. Además, se analizan en detalle los componentes clave del sistema, como el monitor de celdas, el cargador y la pantalla, evaluando su funcionamiento e integración.

Durante el desarrollo se incorporan nuevas funcionalidades. La implementación del balanceo pasivo pone en evidencia fallos en el diseño electrónico de la periferia del circuito bq76PL455A-Q1, que se resuelven mediante ajustes en el hardware. Se implementa además la comunicación con el cargador para ejecutar una curva de carga CC-CV adaptada a los requerimientos de la motocicleta en carrera. Por último, se habilita el control manual de funciones clave mediante la botonera, sin necesidad de reprogramación.

El resultado es la evolución del BMS desde una solución básica de monitorización hacia una gestión completa de baterías para un entorno de competición.

Palabras clave: Motocicleta eléctrica, batería, BMS, bq76PL455A-Q1, balanceo pasivo, CAN bus, carga CC-CV, ESP32.

Abstract: This project presents the development of an efficient battery management system (BMS) for an electric motorcycle designed by the UMA Racing Team, within the framework of the MotoStudent competition. The work builds on an existing electronic platform developed in previous editions and a functional code used solely for monitoring purposes. Based on this foundation, the software is restructured into an object-oriented architecture, extending its capabilities to become a complete BMS with functions for monitoring, balancing, charge control, and protection.

To contextualize the project, a preliminary study is conducted on lithium-ion batteries and BMS technologies, including charge curves and balancing algorithms. Key system components such as the cell monitor, charger, and display are also analyzed in detail, evaluating their operation and integration.

Several new functionalities are incorporated throughout the development. The implementation of passive balancing reveals flaws in the peripheral circuitry around the bq76PL455A-Q1 chip, which are resolved through hardware modifications. Communication with the charger is also implemented, enabling a CC-CV charging profile adapted to the demands of competition. Finally, manual control of key functions is enabled via the button interface, without the need to reprogram the BMS.

The result is the evolution of the system from a basic monitoring solution to a complete battery management system tailored for a competitive environment.

Keywords: Electric motorcycle, battery, BMS, bq76PL455A-Q1, passive balancing, CAN bus, CC-CV charging, ESP32.

Índice

1. Introducción	1
1.1. Antecedentes	1
1.1.1. MotoStudent	1
1.1.2. Batería de la motocicleta eléctrica	4
1.1.3. Trabajos previos sobre el sistema de gestión de baterías	5
1.2. Objetivos del trabajo	6
2. Fundamentos teóricos	9
2.1. Fundamentos de las baterías de ion-litio	9
2.1.1. Construcción y ventajas de las baterías de ion-litio	9
2.1.2. Parámetros y límites de seguridad	9
2.1.2.1. Tensión nominal	10
2.1.2.2. Tensión máxima	10
2.1.2.3. Tensión mínima	10
2.1.2.4. Capacidad	11
2.1.2.5. C-rate, tasa de carga y descarga.	11
2.1.2.6. Corriente máxima de descarga	11
2.1.2.7. Corriente máxima de carga	11
2.1.2.8. Número de ciclos de vida	12
2.1.3. Seguridad	12
2.1.4. Tipos de formato	12
2.1.5. Carga de las baterías	13

2.1.6.	Desbalanceo de celdas	15
2.2.	Fundamentos de un BMS	17
2.2.1.	Introducción al BMS	17
2.2.2.	Clasificación funcional de los BMS	19
2.2.3.	Función de protección en sistemas de gestión de baterías	21
2.2.4.	Función de balanceo en sistemas de gestión de baterías	22
2.2.5.	Estimación del estado de la batería	28
3.	Sistema de gestión eficiente de baterías	31
3.1.	Arquitectura general del sistema	31
3.2.	Descripción general de los módulos	33
3.2.1.	Batería	33
3.2.2.	Red CAN	35
3.2.3.	Hardware BMS	36
3.2.4.	Cargador programable	37
3.2.5.	Pantalla	38
3.2.6.	Controlador	39
3.2.7.	Botonera	40
4.	Hardware BMS	41
4.1.	Arquitectura general del hardware	41
4.2.	ESP32	42
4.2.1.	Integración en el sistema	42
4.2.2.	Comunicación con el sistema	43
4.3.	Monitor de celdas bq76PL455A-Q1	44
4.3.1.	Características técnicas	44
4.3.2.	Comunicación con el sistema	45
4.3.3.	Integración en el sistema	45
4.3.4.	Funciones del bq76PL455A-Q1	46
4.3.4.1.	InitializeBMS()	46
4.3.4.2.	RequestSample()	48

4.3.4.3.	ReadData()	48
4.3.4.4.	balanceCells()	49
4.3.4.5.	RawDataToProcessData()	49
4.3.4.6.	ComunicationCANDisplay()	49
4.3.4.7.	getInfoData()	49
4.3.4.8.	sendInfoData()	49
4.3.4.9.	readOtherBMSData()	50
4.3.4.10.	autobalance()	50
4.4.	Algoritmo de balanceo	50
4.5.	Mejoras implementadas sobre el diseño base	51
5.	Cargador programable	53
5.1.	Características técnicas	53
5.2.	Comunicación con el sistema	54
5.3.	Integración en el sistema	54
5.3.1.	Funciones del Zivan NG3-G7MICB	55
5.3.1.1.	procesarCan()	56
5.3.1.2.	comprobar_conexion()	56
5.3.1.3.	enviar_configuracion_inicial()	56
5.3.1.4.	comprobar_configuracion()	57
5.3.1.5.	enviar_precarga()	57
5.3.1.6.	comprobar_precarga()	58
5.3.1.7.	finalizar_precarga()	58
5.3.1.8.	enviar_porcentaje_corriente()	59
5.3.1.9.	ejecutar_carga_CC()	59
5.3.1.10.	calcular_media_vector()	59
5.3.1.11.	calcular_porcentaje_corriente_CV()	59
5.3.1.12.	ejecutar_carga_CV()	59
5.3.1.13.	detener_cargador()	60
6.	Pantalla	61

6.1. Características técnicas	61
6.2. Comunicación con el sistema	62
6.3. Integración en el sistema	63
7. Implementación software	69
7.1. Funcionamiento general	69
7.2. Inicialización del BMS	71
7.3. Lectura de mensajes CAN	71
7.3.1. Lectura de mensajes CAN del cargador	72
7.4. Control del cargador	75
7.4.1. Estado desconectado del cargador	76
7.4.2. Estado inicio del cargador	76
7.4.3. Estado configuración del cargador	77
7.4.4. Estado precarga del cargador	77
7.4.5. Estado carga_CC del cargador	78
7.4.6. Estado carga_CV del cargador	80
7.4.7. Estado carga_fin del cargador	82
7.5. Balanceo	82
7.6. Monitorización	85
7.7. Protección	89
8. Resultados	91
8.1. Resultados de la monitorización	91
8.2. Resultados del balanceo	93
8.3. Resultados de la carga	96
8.4. Resultados de la protección	98
8.5. Resultados de la interfaz de usuario	99
9. Conclusiones y líneas futuras	101
9.1. Metodología	101
9.2. Logros alcanzados	102

9.3. Limitaciones	103
9.4. Lineas futuras	104
Bibliografía	105

Índice de Tablas

1.	Clasificación de un BMS por funcionalidades	20
1.	Registro CBCONFIG.	47
2.	Registro CHANNELS.	47
3.	Registro NCHAN.	48
4.	Registro CCBENBL.	49
1.	Mensaje CAN 0x380.	56
2.	Mensaje CAN 0x776.	57
3.	Mensaje CAN 0x180.	57
4.	Mensaje CAN 0x600 inicio precarga.	58
5.	Mensaje CAN 0x600 fin precarga.	58
6.	Mensaje CAN 0x6C1.	59
1.	Mensaje CAN 0x300.	64

Índice de Ilustraciones

1.	Logo MotoStudent.	1
2.	Logo MotoStudent electric.	2
3.	Logo UMA Racing Team.	3
4.	Prototipos de UMA Racing Team.	3
5.	Taller 31.	4
6.	Batería de la motocicleta de 2023.	5
1.	Gráfica de la curva de carga CC-CV. Fuente: Yang et al. (2023) [2].	13
2.	Gráfica de la curva de carga multietapa. Fuente: Yang et al. (2023) [2].	14
3.	Gráfica de la curva de carga por pulsos. Fuente: Yang et al. (2023) [2].	15
4.	Coche eléctrico tras incendio del paquete de baterías. Fuente: Andrea (2010) [4].	18
5.	Paquete de baterías de litio de un coche eléctrico tras incendiarse. Fuente: An- drea (2010) [4].	18
6.	Esquema jerárquico para la clasificación funcional de los BMS.	19
7.	Representación esquemática de la función de seguridad de los BMS.	22
8.	Ejemplo práctico del balanceo, estado de la batería inicialmente. Fuente: adap- tado a partir de Andrea (s.f.), Li-Ion BMS [5].	23
9.	Ejemplo práctico del balanceo, estado de la batería con el paso del tiempo. Fuente: adaptado a partir de Andrea (s.f.), Li-Ion BMS [5].	24
10.	Ejemplo práctico del balanceo, carga desbalanceada. Fuente: adaptado a partir de Andrea (s.f.), Li-Ion BMS [5].	25
11.	Ejemplo visual de un algoritmo de balanceo. Fuente: adaptado a partir de Andrea (s.f.), Li-Ion BMS [5].	27

1.	Esquema de las comunicaciones del sistema de gestión de batería.	32
2.	Fotografía del sistema real al completo.	33
3.	Fotografía la batería de la motocicleta.	34
4.	Esquema de las principales conexiones eléctricas de la batería.	34
5.	Estructura de un mensaje can. Extraído de Corrigan (2016) [6]	35
6.	Estructura de una red can. Extraído de Corrigan (2016) [6]	36
7.	Fotografía de los hardware BMS.	37
8.	Fotografía del cargador programable.	38
9.	Fotografía del <i>display</i> AimMXm.	39
10.	Controlador Sevcon gen4. Fuente: [7].	40
11.	Fotografía de la botonera.	40
1.	Modelo 3D del hardware del BMS.	42
2.	Periferia del ESP32 en el hardware del BMS.	42
3.	Modelo 3D de MCP2515T en el hardware del BMS.	43
4.	Periferia del bq76PL455A-Q1 en el hardware del BMS.	44
5.	Modelo 3D de aislador de UART en el hardware del BMS.	45
6.	Electrónica de la placa de desarrollo del bq76PL455A-Q1.	52
7.	Electrónica del hardware del BMS de UMA Racing Team.	52
1.	Conexión CAN del cargador con la motocicleta.	54
2.	Conexión de la potencia del cargador con la motocicleta.	55
1.	Modelo de AimMxM. Fuente: AIM [16].	62
2.	Fragmento del protocolo programado RaceStudio 3.	64
3.	Hoja de visualización de carreras.	65
4.	Hoja de visualización de primeras quince celdas.	66
5.	Hoja de visualización de últimas quince celdas.	66
6.	Hoja de visualización de temperaturas.	67
7.	Hoja de visualización general del BMS.	67
1.	Diagrama general del funcionamiento.	70
2.	Diagrama de inicialización del BMS.	71
3.	Diagrama general de lectura de mensajes CAN.	73

4.	Diagrama de lectura de mensajes CAN del cargador.	74
5.	Diagrama de la máquina de estados del cargador.	75
6.	Diagrama del estado desconectado del cargador.	76
7.	Diagrama del estado inicio del cargador.	76
8.	Diagrama del estado configuración del cargador.	77
9.	Diagrama del estado precarga del cargador.	78
10.	Diagrama del estado carga_CC del cargador.	79
11.	Diagrama de carga_CV del cargador.	81
12.	Diagrama del estado carga_fin del cargador.	82
13.	Diagrama del balanceo del BMS.	83
14.	Diagrama del algoritmo de balanceo.	84
15.	Diagrama de la monitorización del BMS.	86
16.	Diagrama de la gestión de muestra de la monitorización.	88
17.	Diagrama de la protección del BMS.	89
1.	Visualización de las primeras quince celdas.	91
2.	Visualización de las últimas quince celdas.	92
3.	Visualización de la temperatura al inicio del uso.	92
4.	Visualización de la temperatura tras una hora de funcionamiento.	93
5.	Estado inicial de balanceo de las primeras quince celdas.	93
6.	Estado inicial de balanceo de las últimas quince celdas.	94
7.	Balanceo intermedio de las primeras quince celdas.	94
8.	Balanceo intermedio de las últimas quince celdas.	95
9.	Visualización del balanceo en un punto intermedio.	95
10.	Visualización del balanceo final.	96
11.	Visualización del balanceo final.	96
12.	Evolución de la corriente suministrada durante la carga.	97
13.	Evolución de la tensión medida durante la carga.	97
14.	Evolución de la tensión del controlador durante una carga real con perfil CC-CV con eje reescalado.	98

15. Evolución de la tensión del controlador durante una carga real con perfil CC-CV
con eje reescalado. 98

Capítulo 1

Introducción

1.1. Antecedentes

1.1.1. MotoStudent

MotoStudent es una competición internacional promovida por la Fundación Moto Engineering Foundation y TechnoPark MotorLand (Figura 1). Su objetivo es retar a estudiantes universitarios de ingeniería a diseñar, fabricar y validar un prototipo de motocicleta de competición cada dos años, enfrentándose a otros equipos provenientes de universidades de todo el mundo. El desafío combina tanto aspectos técnicos como organizativos y empresariales, ofreciendo una experiencia integral de ingeniería aplicada.



Figura 1. Logo MotoStudent.

La competición se divide en dos categorías: MotoStudent Petrol, en la que los equipos

desarrollan motocicletas con motores de combustión, y MotoStudent Electric, centrada en el diseño de motocicletas formadas exclusivamente por sistemas eléctricos (Figura 2). Esta última representa un escenario idóneo para la investigación, la innovación y el desarrollo de tecnologías sostenibles, en línea con la tendencia global hacia la electrificación del transporte.



Figura 2. Logo MotoStudent electric.

La competición se estructura en dos fases diferenciadas. La primera, denominada MS1, evalúa la propuesta desde un punto de vista industrial, teniendo en cuenta aspectos como el diseño, la viabilidad técnica, la innovación, el control de costes y el plan de negocio. La segunda fase, conocida como MS2, consiste en una serie de pruebas estáticas y dinámicas a los prototipos, incluyendo inspecciones técnicas, ensayos de rendimiento en pista y una carrera final que pone a prueba el comportamiento global del sistema en el circuito de MotorLand Aragón.

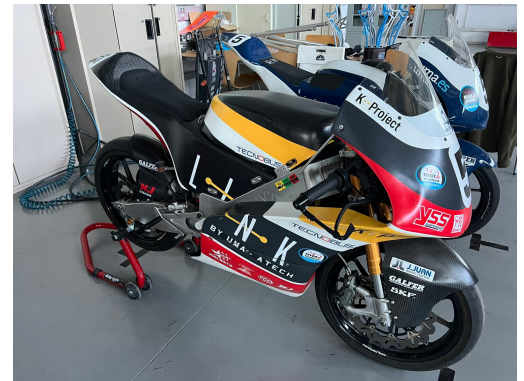
El equipo de la Universidad de Málaga, conocido como UMA Racing Team (Figura 3), participa exclusivamente en la categoría eléctrica con el objetivo de concentrar todos sus esfuerzos en el desarrollo de un prototipo innovador y altamente competitivo (Figura 4.a). Esta elección responde al interés del equipo por enfrentarse a un reto técnico más complejo y menos explorado, que favorece el aprendizaje, la investigación y el desarrollo de soluciones avanzadas en el ámbito de la ingeniería eléctrica y la competición (Figura 4.b). Desde su creación en 2008, el equipo ha estado presente en todas las ediciones de MotoStudent, menos en 2012, recibiendo numerosos premios a lo largo de los años. Entre los más destacados se encuentran el premio Best MotoStudent en 2018, otorgado al equipo con mejor puntuación global en todas las fases de la competición, y el primer premio en pruebas dinámicas en la edición de 2021. Además, en ediciones anteriores a 2016, UMART obtuvo hasta cuatro podios en las categorías de diseño y proyecto.



Figura 3. Logo UMA Racing Team.



(a) Motocicleta de UMA Racing Team en carrera.



(b) Motocicletas de 2021 y 2023.

Figura 4. Prototipos de UMA Racing Team.

El proyecto se desarrolla en el Taller 31 (Figura 5), un entorno multidisciplinar en el que los estudiantes disponen de recursos técnicos y humanos para convertir ideas en soluciones reales. Además del acceso a herramientas, materiales y maquinaria, el equipo cuenta con el apoyo y la supervisión de profesores e investigadores, muchos de ellos con experiencia previa como participantes en MotoStudent.

La complejidad del entorno de competición y las exigencias técnicas de una motocicleta eléctrica hacen necesario el diseño de sistemas inteligentes capaces de gestionar eficazmente el sistema eléctrico del vehículo. En este marco nace el presente Trabajo de Fin de Grado, centrado en el desarrollo de un sistema de gestión eficiente de baterías, pieza clave para garantizar el rendimiento, la seguridad y la operatividad del prototipo en condiciones reales.



(a) Logo del Taller 31.



(b) Nave del Taller 31.

Figura 5. Taller 31.

1.1.2. Batería de la motocicleta eléctrica

En una motocicleta eléctrica de competición, las baterías representan uno de los elementos más críticos del sistema, tanto a nivel funcional como estratégico. No solo son responsables de alimentar el motor y los sistemas electrónicos auxiliares, sino que también condicionan directamente parámetros clave como la autonomía, la aceleración, la estabilidad térmica y el rendimiento en pista. Desde el punto de vista técnico, las baterías constituyen el componente de mayor peso dentro del vehículo y ocupan un volumen significativo, como podemos ver en la Figura 6, donde se marca en rojo el espacio en el que se encuentra la batería en la motocicleta de 2023. La batería influye directamente en la distribución de masas y el diseño estructural del chasis, condiciona el diseño de la motocicleta al completo. Además, su coste económico es elevado, lo que convierte cualquier fallo o degradación prematura en un riesgo importante para el presupuesto del equipo.

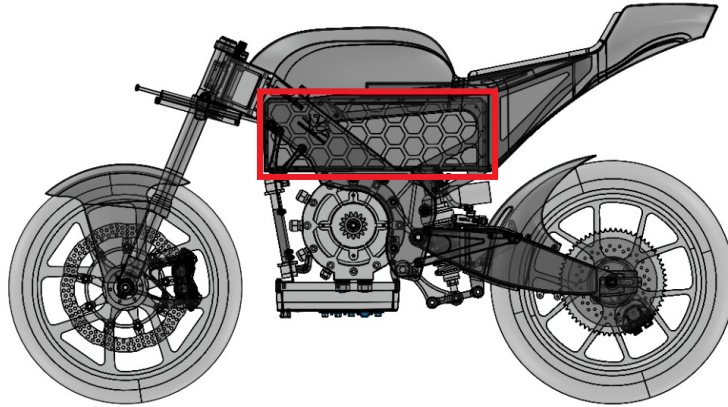


Figura 6. Batería de la motocicleta de 2023.

La importancia de las baterías se acentúa aún más en el entorno competitivo de MotoStudent, donde cada gramo y cada vatio cuentan. Un sistema de gestión ineficiente puede derivar en una pérdida de rendimiento, tiempos de carga excesivos, sobrecalentamientos o desequilibrios entre celdas que comprometen la seguridad. Por el contrario, una batería bien gestionada permite exprimir el potencial del sistema eléctrico sin sobrepasar los límites operativos de seguridad o dañar los componentes.

Por estos motivos, el desarrollo de herramientas que permitan comprender, monitorizar y controlar con precisión el comportamiento de la batería no solo es necesario, sino que constituye una ventaja competitiva tangible. En este proyecto se aborda este reto mediante el desarrollo de un sistema que recopila, procesa y comunica información clave sobre el estado de la batería y su interacción con el resto del vehículo, con el objetivo de optimizar su funcionamiento en todas las fases de uso, desde la carga hasta la conducción.

1.1.3. Trabajos previos sobre el sistema de gestión de baterías

El sistema de gestión de baterías desarrollado en este proyecto no parte desde cero, sino que se apoya en una base técnica construida a lo largo de varias ediciones anteriores del UMA Racing Team. En los últimos años, distintos trabajos de fin de grado han abordado el diseño y puesta en marcha del sistema BMS de la motocicleta, centrándose principalmente en el desarrollo del hardware y en el establecimiento de una comunicación funcional con el chip bq76PL455A-Q1 de Texas Instruments, encargado de medir tensiones y temperaturas de las

celdas.

Estos primeros desarrollos estuvieron orientados a lograr una lectura estable y fiable de los parámetros eléctricos del paquete de baterías. Sin embargo, su alcance se centraba en la adquisición básica de datos, sin llegar a implementar funcionalidades avanzadas como el balanceo de celdas, el control del proceso de carga o la respuesta autónoma ante condiciones de riesgo.

El presente proyecto supone un avance significativo respecto a los trabajos anteriores, al incorporar por primera vez la lógica de balanceo pasivo de celdas, la comunicación automática con el cargador programable, la detección de sobretemperaturas y sobretensiones, así como la monitorización activa de todos estos parámetros. Además, se ha realizado una reestructuración profunda del software, migrando desde un enfoque de programación estructurada hacia un modelo orientado a objetos, lo que mejora notablemente la organización del código, su legibilidad y su escalabilidad para futuras ampliaciones.

Este trabajo consolida así una nueva etapa en la evolución del sistema de gestión de baterías de la motocicleta eléctrica del UMA Racing Team, aportando nuevas funcionalidades y una arquitectura de software mejorada.

1.2. Objetivos del trabajo

El objetivo principal de este Trabajo de Fin de Grado es el desarrollo de un sistema de gestión eficiente de baterías para una motocicleta eléctrica de competición. El sistema debe permitir monitorizar en tiempo real el estado de la batería, coordinar el proceso de carga, ejecutar estrategias de balanceo y comunicar los datos relevantes al resto de módulos de la motocicleta a través del bus CAN.

Para alcanzar este objetivo general, se plantean los siguientes objetivos específicos:

- Reestructurar la aplicación existente en los ESP32 para integrar la adquisición de datos del bq76PL455A-Q1 dentro de una arquitectura orientada a objetos, manteniendo las funcionalidades originales de monitorización y facilitando la escalabilidad del código.
- Implementar un algoritmo de balanceo para igualar el estado de carga entre celdas.

- Establecer comunicación digital con el cargador Zivan NG3 mediante protocolo CAN y controlar su comportamiento durante la carga.
- Detectar condiciones de riesgo, como sobretensión o sobret temperatura, y garantizar una respuesta segura del sistema.
- Diseñar una interfaz de usuario mediante el uso de la botonera de la motocicleta.
- Diseñar un protocolo CAN específico para la motocicleta y una lógica de ejecución no bloqueante que permita que todos los módulos del BMS operen de forma concurrente y sincronizada.

El cumplimiento de estos objetivos permitirá disponer de un sistema robusto, automatizado y funcional, capaz de gestionar de forma eficiente el comportamiento eléctrico de la batería durante todo el proceso operativo de la motocicleta, contribuyendo de forma directa al rendimiento y seguridad del vehículo en el contexto de la competición MotoStudent.

Capítulo 2

Fundamentos teóricos

2.1. Fundamentos de las baterías de ion-litio

A continuación se presenta un estudio sobre el diseño de baterías de ion-litio, basado en la obra de Toll (2017) [1].

2.1.1. Construcción y ventajas de las baterías de ion-litio

Una batería se compone de celdas electroquímicas conectadas en serie y paralelo para alcanzar tensiones y capacidades acordes a las necesidades del sistema. Las celdas conectadas en serie aumentan el voltaje total, mientras que las conexiones en paralelo incrementan la capacidad, lo que permite ajustar la configuración según los requisitos de potencia y autonomía. Las baterías de ion-litio han sido ampliamente adoptadas en aplicaciones de movilidad eléctrica debido a su alta densidad energética y su favorable relación entre peso y capacidad. Estas características permiten construir sistemas compactos, eficientes y ligeros sin sacrificar la autonomía, lo que representa una ventaja clara frente a tecnologías tradicionales como las baterías de plomo-ácido.

2.1.2. Parámetros y límites de seguridad

A la hora de diseñar y gestionar una batería de ion-litio, es imprescindible conocer y respetar una serie de parámetros eléctricos fundamentales. Estos parámetros no solo determinan las características eléctricas del sistema, sino también su comportamiento, su seguridad y

su durabilidad a lo largo del tiempo. Aunque las cifras exactas pueden variar en función del proceso de fabricación específico, los conceptos generales se mantienen constantes. La correcta elección y control de los parámetros que se abordan en este apartado es esencial para garantizar la eficiencia, seguridad y durabilidad del sistema de baterías. El sistema de gestión de baterías debe monitorizar constantemente estos valores para evitar situaciones de riesgo y aumentar la vida útil de la batería.

2.1.2.1. Tensión nominal

Es el voltaje promedio de una celda durante un ciclo de descarga completo. Para las celdas de ion-litio más comunes, la tensión nominal suele situarse alrededor de 3.7 V por celda. Este valor es útil para estimaciones iniciales de diseño, no obstante, no representa ni el voltaje máximo ni el mínimo real que puede alcanzar la celda.

2.1.2.2. Tensión máxima

Corresponde al voltaje al que se considera que una celda está completamente cargada. En la mayoría de las químicas de ion-litio, este valor es de 4.2 V por celda., aunque en algunos casos puede situarse entre los 4.35 V. Superar este límite puede generar una acumulación peligrosa de calor y gases en la celda, lo que incrementa el riesgo de hinchamiento, fuga o incluso explosión. Por este motivo, tanto los cargadores como el BMS deben evitar que ninguna celda exceda esta tensión durante el proceso de carga.

2.1.2.3. Tensión mínima

Es el voltaje por debajo del cual una celda no debe descargarse, ya que hacerlo puede causar una descarga profunda y provocar daños irreversibles. El valor típico de tensión mínima se encuentra sobre los 2.5 V por celda, dependiendo de la química. Hay que tener en cuenta que la curva de descarga de una celda es más pronunciada sobre los 3.4 V, por lo que rápidamente puede atravesar el umbral de mínima tensión si no existen sistemas de monitorización. Si una celda cae por debajo de este umbral, puede sufrir una degradación acelerada de sus materiales y romperse.

2.1.2.4. Capacidad

Se mide en amperios-hora (Ah) e indica la cantidad total de carga eléctrica que una celda puede entregar desde una carga completa hasta el límite de descarga. Por ejemplo, una celda de 2.5 Ah puede suministrar 2.5 amperios durante una hora antes de descargarse. Es uno de los parámetros más importantes a la hora de estimar la autonomía de un sistema eléctrico alimentado por baterías.

2.1.2.5. C-rate, tasa de carga y descarga.

El C-rate describe la relación entre la corriente de carga o descarga y la capacidad nominal de la celda. Nos aporta información sobre la capacidad relativa de la batería de transferir potencia. Por ejemplo, un C-rate de 1C significa que la celda se carga o descarga completamente en una hora. Un C-rate de 2C que se carga o descarga en media hora, y así sucesivamente. El C-rate no es simétrico para la carga y para la descarga.

2.1.2.6. Corriente máxima de descarga

La corriente máxima de descarga proporciona información sobre la carga máxima que soporta la batería. Es la corriente máxima que la batería puede aportar sin sobrecalentarse o dañarse en una descarga continuada en régimen estacionario. Los fabricantes proporcionan adicionalmente un parámetro de corriente pico máximo de descarga. Que es la cantidad de corriente que una celda puede proporcionar en el plazo de unos pocos segundos. En configuraciones de baterías para motocicletas eléctricas de competición, estos parámetros deben elegirse cuidadosamente, ya que influye en la capacidad de respuesta del sistema ante aceleraciones rápidas.

2.1.2.7. Corriente máxima de carga

La corriente máxima de carga se expresa con el C-rate y suelen ser valores inferiores al C-rate de descarga. La mayoría de los fabricantes recomiendan no superar un C-rate de 0.5C, puesto que valores altos de C de carga pueden acortar la vida útil de la batería. Se suele reducir el C de carga a 0.2 C para aumentar al máximo el número de ciclos de vida de la batería.

2.1.2.8. Número de ciclos de vida

Representa la cantidad de ciclos completos de carga-descarga que una celda puede soportar antes de que su capacidad se degrade por debajo de un umbral, comúnmente el ochenta por ciento de su valor nominal. Este número puede variar considerablemente según la calidad de la celda, su química, el perfil de uso y la gestión térmica.

2.1.3. Seguridad

Las baterías de ion-litio, aunque ampliamente utilizadas por su eficiencia y rendimiento, presentan riesgos que deben gestionarse adecuadamente. Uno de los principales peligros es el cortocircuito interno o externo, que puede desencadenar un aumento repentino de temperatura y suponen un gran riesgo de incendio o explosión. Otro factor crítico es la temperatura de la batería. Operar fuera del rango seguro acelera la degradación de las celdas y puede producir que la batería se incendie. Además, la manipulación inadecuada durante el montaje o mantenimiento del sistema puede dañar físicamente las celdas, generar contactos inseguros o afectar a los sistemas de protección. Para una motocicleta de competición es fundamental realizar un montaje seguro con una protección adecuada para prevenir a la batería de impactos o desplazamientos que supongan un riesgo para la seguridad del piloto y del vehículo.

2.1.4. Tipos de formato

Las baterías de ion-litio se comercializan en distintos formatos que presentan distintas ventajas o desventajas. Las celdas de tipo *pouch* o bolsa son celdas planas, muy eficientes en cuanto a volumen ocupado, pero no tienen ninguna protección, por lo que es necesario incluir algún tipo de carcasa o seguridad en las aplicaciones. Las celdas prismáticas son celdas planas que contienen una carcasa protectora, por lo que son menos eficientes en espacio, pero no requieren ningún tipo de carcasa. Las celdas cilíndricas son celdas con formato de pilas comerciales, son las menos eficientes en el espacio, pero las más resistentes. Estas además tienen un estándar

2.1.5. Carga de las baterías

Este apartado se basa en los principios y perfiles de carga descritos por Yang et al. (2023) [2], que analizan distintos métodos y estrategias adaptadas a las celdas de ion-litio.

El proceso de recarga de una batería de ion-litio puede seguir diferentes perfiles según las necesidades de la aplicación y las características del sistema energético en el que se integra estos perfiles están diseñados para maximizar la seguridad la eficiencia y la vida útil de la batería teniendo en cuenta factores como la temperatura el envejecimiento y el estado de carga inicial de las celdas.

El perfil de carga más habitual y ampliamente aceptado en aplicaciones industriales es el método de corriente constante y tensión constante (CC-CV) (Figura 1). En la primera fase del proceso se suministra una corriente constante que permite cargar rápidamente la batería mientras su tensión va aumentando gradualmente. Cuando se alcanza la tensión objetivo, normalmente la tensión máxima establecida por el fabricante, se pasa a una segunda fase en la que la tensión se mantiene fija mientras la corriente disminuye progresivamente hasta alcanzar un valor mínimo de corte. Este método garantiza una carga completa sin sobrepasar los límites seguros de tensión ni de corriente, lo que lo convierte en el estándar en sistemas de movilidad eléctrica.

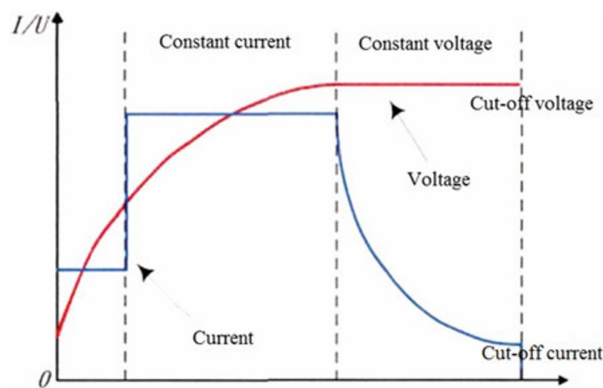


Figura 1. Gráfica de la curva de carga CC-CV. Fuente: Yang et al. (2023) [2].

Otro perfil común es la carga por etapas o multietapa (2) que consiste en aplicar distintos niveles de corriente constante en función del estado de la batería a medida que la tensión se

incrementa se reduce la corriente en pasos para adaptar el proceso al comportamiento interno de las celdas este enfoque puede ser útil en sistemas que presentan una alta sensibilidad térmica o una elevada variabilidad entre celdas aunque en general presenta una menor eficiencia temporal y una implementación más compleja que el perfil CC-CV.

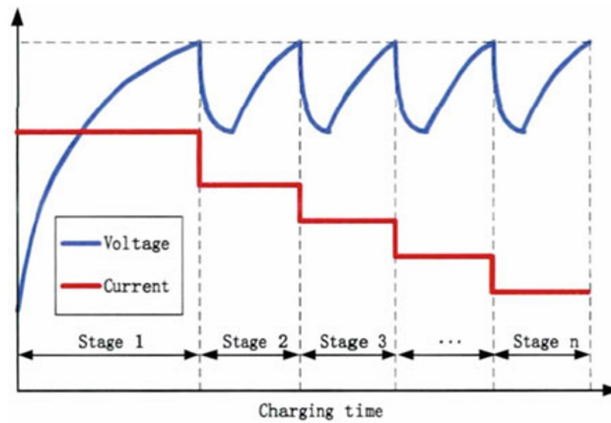


Figura 2. Gráfica de la curva de carga multietapa. Fuente: Yang et al. (2023) [2].

También se emplea el perfil de carga por pulsos o pulse charging (3) el cual alterna periodos de corriente constante con pausas sin suministro de corriente durante estas pausas se produce un efecto de relajación que favorece el reequilibrio electroquímico en el interior de las celdas este mecanismo contribuye a reducir la temperatura de funcionamiento mejorar la distribución interna de carga y aumentar tanto la capacidad efectiva como la vida útil de la batería este perfil resulta especialmente interesante en contextos donde se requiere minimizar el calentamiento y acelerar la carga sin comprometer la seguridad del sistema.

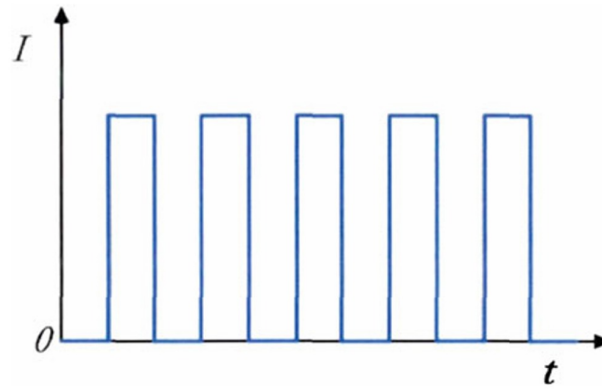


Figura 3. Gráfica de la curva de carga por pulsos. Fuente: Yang et al. (2023) [2].

Existen además perfiles de carga optimizados mediante algoritmos avanzados que permiten adaptar dinámicamente la corriente y la tensión en función del estado interno de la batería. Uno de los enfoques más estudiados es el basado en un algoritmo de colonia de hormigas, propuesto por Liu, Teng y Lin (2005) [3], que busca maximizar la capacidad final cargada mediante técnicas inspiradas en la naturaleza. Sin embargo, el beneficio práctico de estos algoritmos es limitado en comparación con la complejidad que implican.

2.1.6. Desbalanceo de celdas

En la fabricación de baterías es habitual la necesidad de asociar múltiples celdas en paralelo y en serie para alcanzar los niveles de tensión y capacidad requeridos por la aplicación. Cualquier configuración de celdas que incluya celdas en serie introduce un fenómeno conocido como desbalanceo de celdas, el cual se manifiesta como diferencias de tensión entre las distintas celdas que forman parte del paquete.

Según la obra de Andrea (2010) [4], el desbalanceo se origina tanto por factores internos como externos. A nivel interno, cada celda presenta ligeras variaciones de capacidad, resistencia interna y eficiencia de carga debido a tolerancias de fabricación. A nivel externo, influyen factores como la distribución de temperatura dentro del paquete, diferencias en la resistencia de conexión o asimetrías en el diseño del sistema. Estas pequeñas diferencias no tienen un impacto significativo durante los primeros ciclos de uso, pero con el paso del tiempo y la repetición de múltiples ciclos de carga y descarga, las diferencias entre celdas se acentúan

progresivamente.

Por ejemplo, supongamos una batería compuesta por cuatro celdas en serie inicialmente iguales, todas con una tensión de 4.00 V y en conjunto con una tensión total de 16.00V. Tras numerosos ciclos, podrían observarse tensiones de 3.95 V en una celda y 4.05 V en otra, aunque la suma total siga siendo 16.00V parece que todo está funcionando correctamente, pero existe un desbalanceo. Sin una monitorización individual, el sistema percibiría la tensión global como normal, sin detectar que una de las celdas ha sobrepasado el umbral esperado de 4.00V. Esto puede llevar a que, durante una carga, una celda supere su tensión máxima segura, con el riesgo de incendio o explosión. En el extremo opuesto, una celda desbalanceada podría alcanzar una tensión por debajo del umbral mínimo durante la descarga, provocando daños irreversibles en su estructura interna.

Además de las diferencias internas en capacidad, resistencia o eficiencia durante los ciclos de carga y descarga, otro fenómeno que puede provocar el desbalanceo entre celdas es la autodescarga. Aunque las celdas de ion-litio presentan una autodescarga muy baja comparada con otras tecnologías, esta no es exactamente igual entre todas las celdas. Con el tiempo, pequeñas diferencias en la corriente de fuga interna o en las pérdidas por el propio sistema de gestión de baterías pueden hacer que algunas celdas pierdan más carga que otras incluso cuando el paquete está inactivo. Este tipo de desbalanceo, que se acumula lentamente durante periodos prolongados sin uso, puede afectar negativamente a la capacidad disponible y al equilibrio general del sistema si no se corrige mediante un apropiado balanceo.

Estos fenómenos son particularmente problemáticos en configuraciones en serie, ya que las celdas no pueden corregirse entre sí de forma natural. A diferencia de las asociaciones en paralelo, donde las celdas tienden a equilibrarse automáticamente al compartir la misma tensión. Por tanto, es fundamental tener en cuenta el desbalanceo de celdas en el diseño y operación de sistemas de baterías. La monitorización continua del estado individual de cada celda, así como la implementación de mecanismos de balanceo, resultan esenciales para garantizar la seguridad, eficiencia y durabilidad del sistema.

2.2. Fundamentos de un BMS

En esta sección se abordan en profundidad los principios de funcionamiento, arquitectura y funciones esenciales de un sistema de gestión de baterías orientado a aplicaciones de potencia según la obra de Andrea (2010) [4].

2.2.1. Introducción al BMS

Un sistema de gestión de baterías, o Battery Management System (BMS), es un dispositivo electrónico que se integra en un sistema energético con baterías recargables y cuya misión principal es garantizar su funcionamiento seguro, eficiente y fiable. Se considera BMS a cualquier sistema capaz de cumplir una o varias de las siguientes funciones: monitorizar el estado de la batería, protegerla frente a condiciones de peligro, estimar parámetros internos como el estado de carga o de salud, aumentar la eficiencia de la batería o comunicar información clave a sistemas externos o al usuario.

En el contexto de vehículos eléctricos, especialmente aquellos destinados a competición, la presencia de un BMS no solo es recomendable, sino imprescindible. Las baterías utilizadas en este tipo de aplicaciones presentan una alta densidad energética, tienen un coste elevado y están sujetas a condiciones de operación muy exigentes. Además, cualquier desviación de los parámetros eléctricos o térmicos recomendados puede derivar en situaciones de riesgo, tanto para el vehículo como para las personas. Tal como se ilustra en las Figuras 4 y 5, una batería de litio puede incendiarse o explotar si se exceden ciertos límites de operación generando grandes daños y riesgos.



Figura 4. Coche eléctrico tras incendio del paquete de baterías. Fuente: Andrea (2010) [4].



Figura 5. Paquete de baterías de litio de un coche eléctrico tras incendiarse. Fuente: Andrea (2010) [4].

Por este motivo, es responsabilidad del BMS mantener la batería dentro de lo que se denomina área segura de operación, o *Safe Operating Area* (SOA). Este dominio, definido por parámetros como tensión, corriente y temperatura, delimita las condiciones en las que la batería puede operar sin comprometer su integridad. La capacidad del BMS para supervisar este estado en tiempo real y prevenir desviaciones resulta esencial en aplicaciones de alta exigencia como la competición.

2.2.2. Clasificación funcional de los BMS

Existen múltiples formas de clasificar los sistemas de gestión de baterías en función de su arquitectura, grado de integración o aplicación específica. Sin embargo, una de las clasificaciones más claras y útiles desde el punto de vista técnico es la que se basa en la funcionalidad que cada sistema es capaz de ofrecer. Esta categorización, propuesta por Davide Andrea en su obra sobre sistemas de gestión de baterías de ion-litio, ordena los BMS según el nivel de complejidad y capacidad de intervención, desde los sistemas más simples hasta los más completos, como se puede ver en la Figura 6.

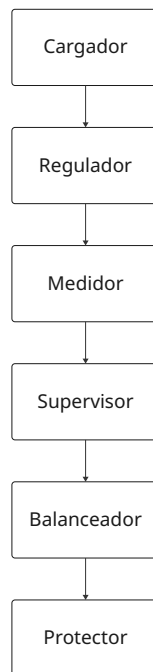


Figura 6. Esquema jerárquico para la clasificación funcional de los BMS.

En el nivel más básico se encuentran los sistemas que simplemente actúan como cargadores sin realizar ninguna función adicional. Estos dispositivos siguen una lógica de corriente constante y tensión constante, y no incorporan ningún tipo de supervisión o comunicación con el exterior. Su simplicidad los hace aptos únicamente para aplicaciones donde la seguridad no depende de una supervisión precisa.

Un paso por encima se encuentran los reguladores, que son capaces de limitar la tensión o la corriente mediante circuitos de protección o elementos de corte. Aunque pueden evitar daños en condiciones extremas, carecen de capacidad de monitorización continua o de recolección de datos, por lo que su intervención se basa en umbrales fijos.

Los medidores son sistemas que incorporan funciones de medición sin capacidad de actuación. Se limitan a registrar variables como la tensión, la temperatura o la corriente, y pueden transmitir esta información a otros dispositivos del sistema o almacenarla para análisis posteriores. Aunque son útiles para tareas de diagnóstico, visualización o registro de datos, no participan activamente en el control ni en la protección del sistema de baterías.

A partir de ahí, aparecen los supervisores, que combinan la medición con la detección de situaciones anómalas. Estos sistemas pueden generar alertas, detener procesos o desconectar el sistema si detectan condiciones de operación peligrosas, aunque su capacidad de intervención suele estar limitada a acciones predeterminadas.

Los balanceadores añaden la capacidad de corregir las diferencias de carga entre celdas. Esto permite mantener un estado de carga homogéneo en todo el paquete, prolongando su vida útil y mejorando el rendimiento. El balanceo puede realizarse mediante técnicas pasivas, disipando energía, o mediante técnicas activas, transfiriéndola entre celdas.

Finalmente los protectores representan el nivel más avanzado de funcionalidad. Además de incorporar medición, supervisión, comunicación y balanceo, integran capacidades de protección activa ante condiciones críticas y control dinámico del sistema de carga y descarga.

Tipo de BMS	Carga	Protección	Medicion	Balanceo
Cargador	Básica CC-CV	No	No	No
Regulador	Básica CC-CV	Elementos pasivos	No	No
Medidor	Básica CC-CV	No	Si	No
Supervisor	Control amplio	Indirecta	Si	No
Balanceador	Control amplio	Indirecta	Si	Si
Protector	Control amplio	Activa	Si	Si

Tabla 1. Clasificación de un BMS por funcionalidades

Esta clasificación (Tabla 1) no depende directamente de la arquitectura o del tipo de

implementación del sistema, sino que permite situar cada diseño en función del nivel de supervisión, control y capacidad de actuación que ofrece. Resulta especialmente útil para contextualizar el alcance funcional de una solución concreta y valorar su grado de intervención dentro del sistema en el que se integra. En este proyecto se ha buscado desarrollar un sistema completo, capaz de monitorizar, balancear, proteger y controlar la carga de la batería, lo que lo sitúa dentro de la categoría de BMS protector según esta clasificación.

2.2.3. Función de protección en sistemas de gestión de baterías

En sistemas de baterías avanzados, como los utilizados en motocicletas eléctricas de competición, el BMS desempeña un papel esencial en la protección activa del sistema. Su función va más allá de la simple lectura de parámetros eléctricos, asumiendo la responsabilidad de detectar y responder a condiciones que puedan comprometer la integridad de la batería o la seguridad del conjunto.

La protección se basa en la supervisión continua de variables críticas como la tensión, la corriente y la temperatura. El sistema compara en tiempo real estos valores con los márgenes definidos por el fabricante de las celdas o por los requerimientos de la aplicación, y actúa en consecuencia cuando detecta una desviación potencialmente peligrosa. Estas acciones pueden incluir la reducción de la corriente de carga o descarga para controlar la temperatura, o bien la activación de sistemas de refrigeración si el diseño del vehículo los contempla.

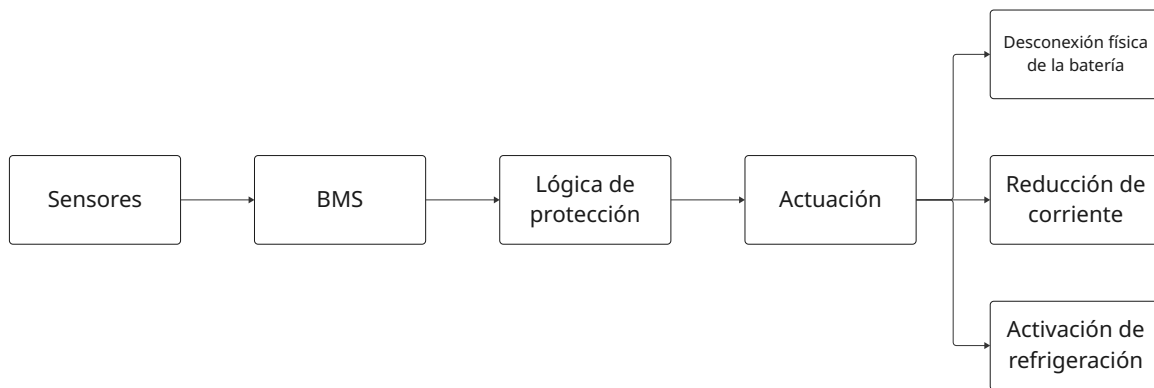


Figura 7. Representación esquemática de la función de seguridad de los BMS.

El objetivo de esta lógica de protección es mantener siempre la batería operando dentro de su área segura, previniendo situaciones que puedan reducir su vida útil, dañar sus componentes internos o generar riesgos térmicos graves.

En algunas arquitecturas, la actuación física ante condiciones críticas, como la desconexión del sistema o la interrupción del flujo de corriente, queda delegada en módulos externos como el controlador o un sistema de contactores. No obstante, existen sistemas en los que el propio BMS dispone de una salida de control dedicada para accionar directamente un contactor de seguridad, lo que le permite cortar la conexión eléctrica de forma inmediata y autónoma ante la detección de condiciones peligrosas. Esta capacidad de actuación directa refuerza el papel del BMS como componente central en la estrategia de protección del sistema, especialmente en entornos donde la velocidad de respuesta y la independencia funcional son prioritarias.

2.2.4. Función de balanceo en sistemas de gestión de baterías

El balanceo es una de las funciones clave que puede incorporar un sistema de gestión de baterías, especialmente en configuraciones donde las celdas están conectadas en serie. Como se explicó anteriormente, el desbalanceo es un fenómeno inevitable que surge con el uso, debido a diferencias internas entre las celdas; o debido a la autodescarga. Esta falta de uniformidad

provoca que, con el tiempo, algunas celdas alcancen tensiones superiores o inferiores a otras, lo que reduce la capacidad útil del sistema y puede comprometer su seguridad. El balanceo busca corregir estas desviaciones, manteniendo las tensiones de las celdas lo más alineadas posible.

El balanceo de celdas no solo tiene una justificación teórica, sino que puede ilustrarse de forma clara con un ejemplo práctico. Supongamos una batería compuesta por cuatro celdas conectadas en serie. Al inicio de su vida útil, estas celdas tienen la misma cantidad de carga, como se muestra en la Figura 8. En esta situación el sistema puede aprovechar plenamente la capacidad de carga y descarga de la batería sin comprometer su integridad.

Las figuras 8 a 11 han sido generadas a partir del contenido visual interactivo publicado por Andrea (s.f.) en su sitio web Li-Ion BMS [5]. Aunque no se han extraído directamente, se han construido a partir de los datos y dinámicas presentadas en el gráfico, adaptándolos al sistema desarrollado.

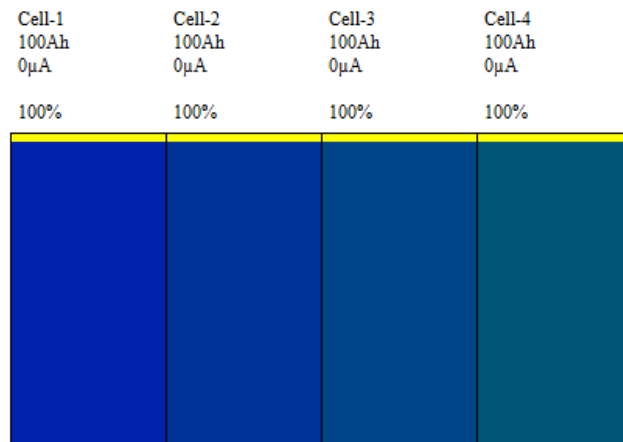


Figura 8. Ejemplo práctico del balanceo, estado de la batería inicialmente.
Fuente: adaptado a partir de Andrea (s.f.), Li-Ion BMS [5].

Sin embargo, tras numerosos ciclos de uso y periodos de inactividad, comienzan a manifestarse diferencias entre las celdas. Como consecuencia, las tensiones individuales se desalinean, como puede observarse en la Figura 9. Aunque la tensión total del paquete pueda seguir pareciendo normal, algunas celdas se encontrarán más cerca de sus límites máximos o mínimos que otras.

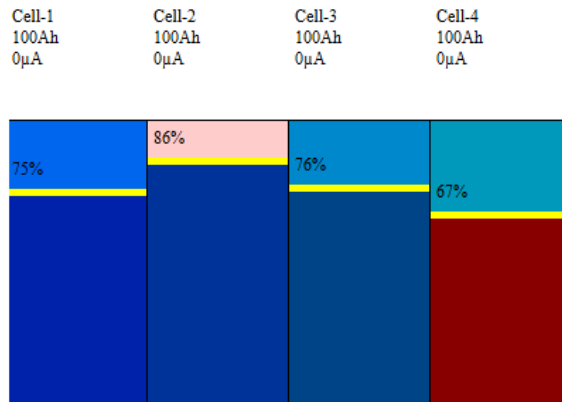


Figura 9. Ejemplo práctico del balanceo, estado de la batería con el paso del tiempo. Fuente: adaptado a partir de Andrea (s.f.), Li-Ion BMS [5].

En este estado de desbalanceo, si se intenta cargar la batería sin realizar ningún tipo de compensación, el proceso debe detenerse tan pronto como la celda con mayor tensión alcance su límite superior. Esto obliga a interrumpir la carga antes de que las demás celdas se carguen completamente. El resultado es una pérdida efectiva de capacidad y tensiones utilizables, como se representa en la Figura 10. En casos extremos, si no se supervisan las celdas de forma individual y se carga el paquete solo en función de su tensión global, se corre el riesgo de sobrecargar una celda o de descargarla por debajo de su umbral seguro, lo que puede desencadenar en daños irreversibles o eventos peligrosos.

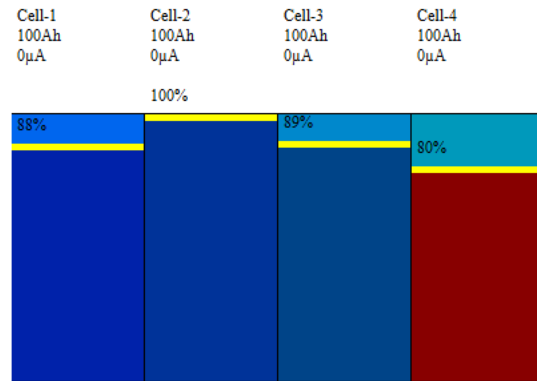


Figura 10. Ejemplo práctico del balanceo, carga desbalanceada. Fuente: adaptado a partir de Andrea (s.f.), Li-Ion BMS [5].

Este ejemplo evidencia la importancia de incorporar mecanismos de balanceo en el sistema de gestión. El balanceo permite igualar las tensiones de las celdas antes o durante el proceso de carga, maximizando la capacidad disponible y evitando situaciones de riesgo. Además, destaca la necesidad de contar con un BMS capaz de monitorizar cada celda individualmente, ya que una supervisión global del paquete no es suficiente para garantizar la seguridad ni el rendimiento a largo plazo del sistema.

El proceso de balanceo puede abordarse desde diferentes estrategias, y su implementación está condicionada por el tipo de hardware disponible y los objetivos del sistema. Uno de los enfoques más relevantes es el diseño del algoritmo de activación del balanceo. Existen métodos basados en el historial del estado de carga de las celdas, aunque en la práctica, los más comunes se fundamentan en el voltaje medido de las celdas, especialmente cuando este se aproxima al final del proceso de carga. En esta zona, la correlación entre voltaje y nivel de carga es más precisa, lo que permite decidir con mayor fiabilidad cuándo y sobre qué celdas aplicar el balanceo.

El balanceo puede clasificarse principalmente en dos enfoques: pasivo y activo. Ambos tienen el objetivo de corregir las diferencias de carga entre celdas pero se diferencian en su forma de hacerlo, así como en sus técnicas y diseño.

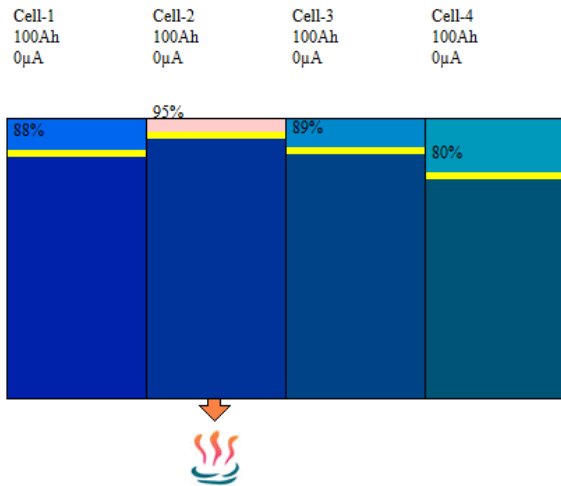
En el balanceo pasivo, la energía excedente de las celdas más cargadas se disipa en forma

de calor mediante resistencias de descarga. Es una técnica sencilla, robusta y económica, lo que la hace muy utilizada en sistemas donde se prioriza la fiabilidad y la integración. Aunque implica una pérdida energética directa, esta es despreciable en comparación con la energía total almacenada en la batería, por lo que su impacto en el rendimiento global del sistema es muy reducido. Una desventaja de este enfoque es que, en muchas soluciones integradas en chip, las corrientes de balanceo están limitadas, lo que puede aumentar considerablemente el tiempo de balanceo si las diferencias entre celdas son elevadas.

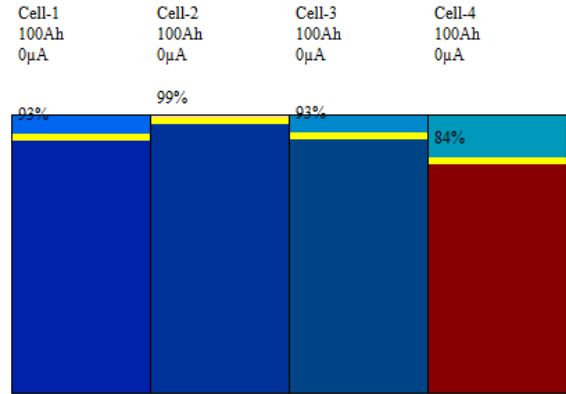
Para ilustrar de forma visual el funcionamiento del balanceo pasivo, retomamos el ejemplo mostrado en la Figura 10, donde una de las celdas ha alcanzado el 100% de su capacidad mientras las demás se mantienen en niveles inferiores. En este punto, el proceso de carga se detiene para evitar que la celda más cargada supere su tensión máxima de seguridad.

A continuación, se activa un algoritmo de balanceo pasivo basado en voltajes. Este algoritmo identifica qué celdas superan un umbral predefinido, por ejemplo, el 95% de su tensión máxima permitida y activa resistencias de balanceo sobre ellas para reducir con el tiempo su tensión. El objetivo es que todas las celdas estén por debajo de este umbral antes de reanudar el proceso de carga. Este ciclo de balanceo y carga se repite de forma iterativa hasta que las celdas alcanzan tensiones muy similares.

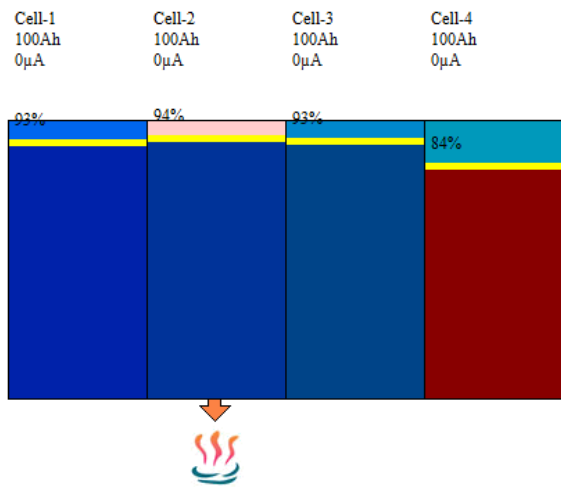
En la Figura 11.a se muestra la primera iteración del proceso: la celda 2, que presenta la mayor tensión relativa, es balanceada mediante la activación de una resistencia, representada mediante un símbolo bajo la celda. Una vez su tensión desciende por debajo del umbral del 95%, se reinicia la carga del paquete, como se ilustra en la Figura 11.b. Sin embargo, al seguir existiendo diferencias, el algoritmo vuelve a detectar desequilibrio y repite el ciclo. En la Figura 11.c se vuelve a descargar parcialmente la celda 2, y en la Figura 11.d se reanuda la carga. A medida que se reduce el desbalance, más celdas alcanzan el umbral superior: en la Figura 11.e, las celdas 1, 2 y 3 se balancean simultáneamente. Finalmente, en la Figura 11.f se alcanza una situación de equilibrio en la que todas las celdas se encuentran próximas al 100% de su capacidad y la batería puede considerarse completamente cargada de forma segura y homogénea.



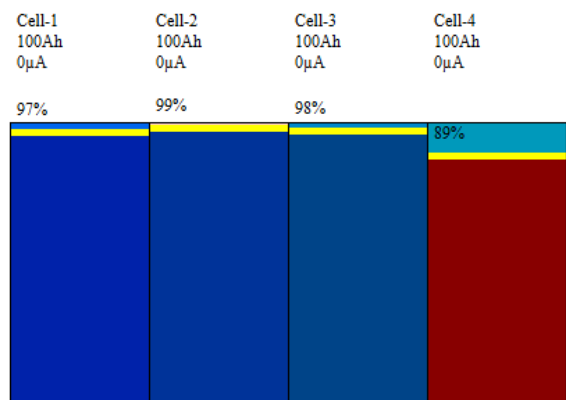
(a) Primera descarga.



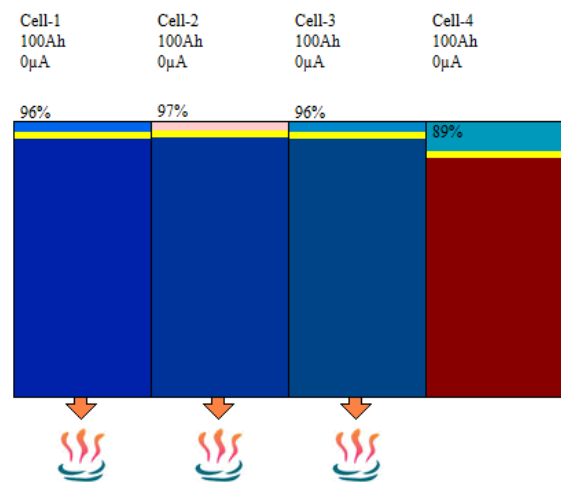
(b) Primera carga.



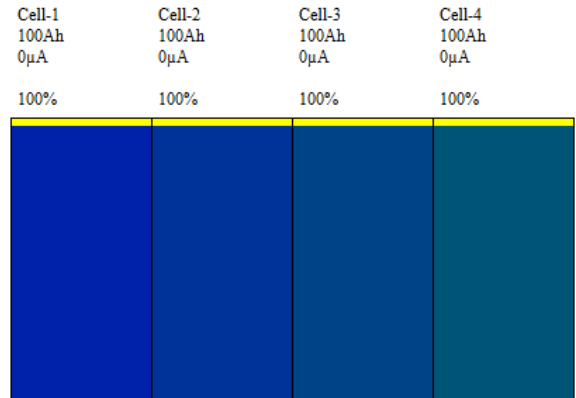
(c) Segunda descarga.



(d) Segunda carga.



(e) Tercera descarga.



(f) Fin del algoritmo.

En el balanceo activo, en lugar de disipar la energía sobrante, se transfiere desde las celdas más cargadas hacia las menos cargadas mediante elementos activos. La transferencia de la energía puede ser entre celdas adyacentes, de celda a batería o de batería a celda. Este método mejora la eficiencia al conservar la energía dentro del sistema, pero requiere un diseño más complejo, con componentes adicionales, mayor volumen y mayor coste. Aunque ofrece ventajas teóricas en cuanto a aprovechamiento energético, en la práctica no suele utilizarse en sistemas compactos o de competición, donde las restricciones de espacio, fiabilidad y sencillez de mantenimiento tienen mayor peso.

Existe también una alternativa al uso de un BMS con función de balanceo, basado en el uso de un cargador distribuido. En esta configuración, cada celda o grupo de celdas dispone de un cargador independiente que se encarga de controlar su carga. Este método elimina el riesgo de sobrecargar una celda, pero implica una complejidad y un tiempo de carga considerablemente mayores, lo que lo hace poco viable para sistemas de alto rendimiento o vehículos eléctricos.

La elección del tipo de balanceo no debe basarse únicamente en criterios de eficiencia energética, sino también en factores previamente mencionados, como la arquitectura del sistema, la criticidad de la aplicación y las restricciones de espacio y coste. Por ello, muchos BMS comerciales y de competición optan por soluciones pasivas integradas, suficientemente efectivas para mantener el paquete en condiciones óptimas sin añadir complejidad innecesaria.

2.2.5. Estimación del estado de la batería

Además de las funciones de protección y balanceo, algunos sistemas de gestión de baterías incorporan algoritmos para la estimación del estado interno de la batería. Entre los más relevantes se encuentran el *State of Charge* (SOC), que representa el nivel de carga disponible de la batería en tiempo real, y el *State of Health* (SOH), que cuantifica la degradación de la batería en comparación con sus condiciones nominales de fabricación. Ambos parámetros son fundamentales en aplicaciones donde se requiere conocer con precisión la autonomía restante o anticipar el fin de vida útil del sistema.

A diferencia de las variables físicas que el BMS puede medir directamente, como tensión, corriente o temperatura, el SOC y el SOH son valores que deben estimarse mediante modelos matemáticos o técnicas complejas de filtrado. Estas técnicas requieren una caracterización

previa del sistema, sensores de alta precisión y, en muchos casos, capacidad de cómputo adicional. Además, factores como la temperatura, la tasa de descarga o el envejecimiento no lineal complican su fiabilidad.

Por este motivo, en sistemas como el desarrollado en este proyecto, orientados a la robustez y la funcionalidad, no se han implementado algoritmos avanzados de estimación del SOC o el SOH. Los requerimientos operativos del vehículo se satisfacen adecuadamente mediante la monitorización directa de los parámetros eléctricos fundamentales.

Capítulo 3

Sistema de gestión eficiente de baterías

3.1. Arquitectura general del sistema

El sistema de gestión eficiente de baterías desarrollado en este proyecto se concibe como un conjunto de funciones distribuidas dentro de la electrónica de la motocicleta. No se trata de un único dispositivo o placa, sino de un sistema capaz de recopilar información de distintos módulos, procesarla y actuar en consecuencia para controlar el estado de la batería de la motocicleta. Este enfoque busca automatizar tareas como la monitorización, el balanceo de celdas, la coordinación de la carga o la detección de condiciones anómalas, mejorando así la seguridad y el rendimiento del conjunto.

Cuando a lo largo de este trabajo se utilice el término BMS, podrán distinguirse dos interpretaciones según el contexto. Desde una perspectiva funcional y abstracta, el BMS hace referencia al conjunto de software y lógica de control que permite supervisar el estado de la batería, coordinar el proceso de carga y actuar sobre el sistema en función de las condiciones detectadas. En cambio, desde un punto de vista práctico y físico, se utilizará también el término BMS para referirse a cada placa electrónica que integra el chip bq76PL455A-Q1 y el microcontrolador ESP32 sobre el que se ejecuta dicha lógica. Esta dualidad responde tanto a la necesidad de distinguir claramente el enfoque conceptual como a la terminología habitual empleada en el entorno de desarrollo del equipo.

La lógica del sistema se implementa en dos microcontroladores ESP32, cada uno de ellos integrado en una de las dos placas BMS de la motocicleta. Ambos microcontroladores adquieren datos de tensión y temperatura a través del chip bq76PL455A-Q1 presente en su misma PCB, y se comunican entre sí mediante el bus CAN para compartir la información completa del conjunto de celdas. Esta información se transmite también a la pantalla o *display*, lo que permite al piloto visualizar en tiempo real el estado del sistema. Tras el intercambio de datos, cada microcontrolador ejecuta de forma independiente el balanceo pasivo sobre su grupo de celdas asignadas. Uno de los dos ESP32 se configura como maestro, asumiendo funciones adicionales: controla el proceso de carga a través del cargador, interactúa con el controlador del motor y recibe señales desde la botonera, lo que le permite tomar decisiones relacionadas con la seguridad, la carga y descarga de la batería.

En la Figura 1 se presenta un esquema de la arquitectura de comunicaciones general del sistema, donde se representan los módulos involucrados. La Figura 2 complementa esta vista con una fotografía de la motocicleta real, en la que se han señalado en recuadros rojos los distintos elementos del sistema.

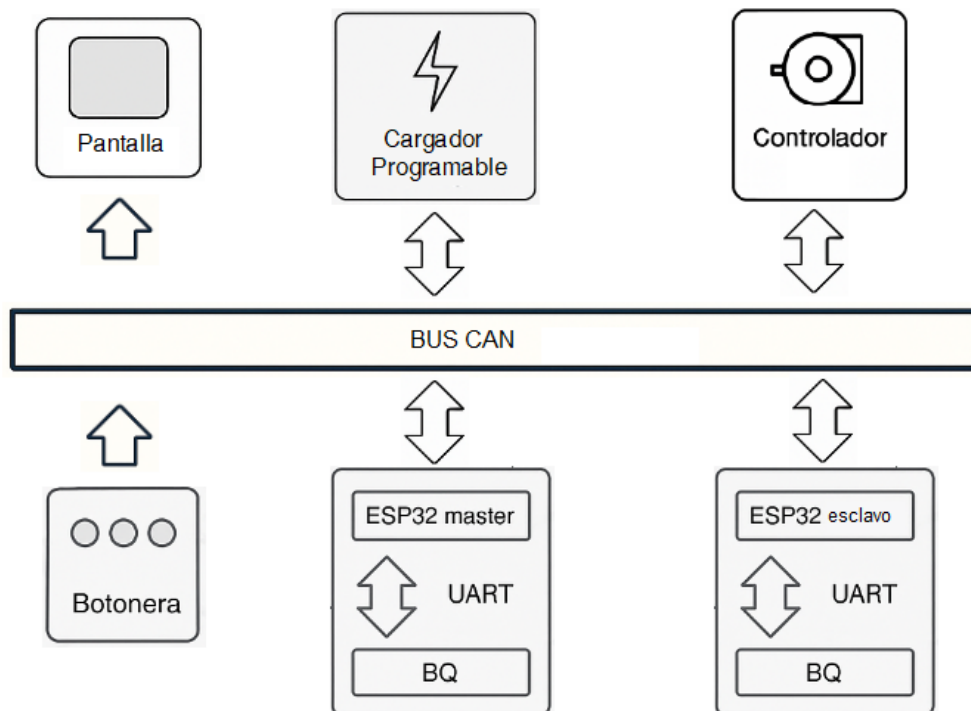


Figura 1. Esquema de las comunicaciones del sistema de gestión de batería.

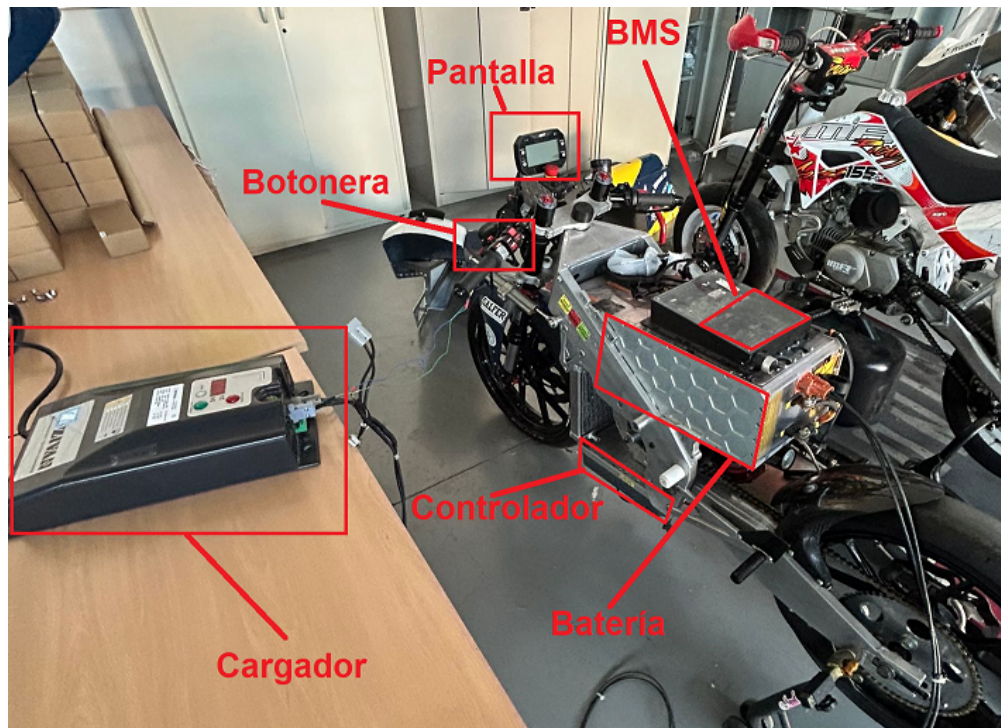


Figura 2. Fotografía del sistema real al completo.

3.2. Descripción general de los módulos

3.2.1. Batería

La batería (Figura 3) es el componente encargado de almacenar y suministrar la energía necesaria para alimentar tanto el motor eléctrico como el resto de módulos del sistema. Está compuesta por un conjunto de celdas de ion-litio conectadas en serie y paralelo, formando un paquete de alta tensión diseñado específicamente para las exigencias de la competición.



Figura 3. Fotografía la batería de la motocicleta.

Contiene tres conexiones principales: la conexión de carga, que enlaza con el cargador programable; la conexión de descarga, dirigida al controlador; y las líneas de monitorización, que permiten al sistema BMS adquirir datos de tensión y temperatura de las celdas. Esta información es clave para coordinar funciones como el balanceo y la gestión de carga, garantizando la seguridad y el rendimiento del vehículo. El esquema de estas conexiones se puede ver en la Figura 4.

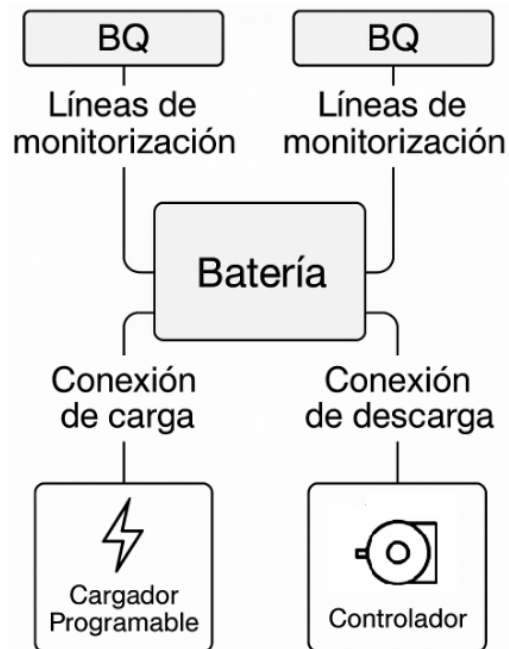


Figura 4. Esquema de las principales conexiones eléctricas de la batería.

Las medidas de tensión y temperatura de cada celda o grupo de celdas se recogen a través de líneas de monitorización conectadas a los chips bq76PL455A-Q1. Esta información es fundamental para mantener el equilibrio del sistema, detectar posibles fallos y optimizar el rendimiento durante la conducción y la recarga. La batería, por tanto, es el componente crítico del sistema eléctrico cuya gestión eficiente resulta crucial para garantizar el correcto funcionamiento del vehículo.

3.2.2. Red CAN

En esta motocicleta se utiliza una red CAN para la comunicación entre los distintos módulos del sistema. El protocolo CAN (*Controller Area Network*) es ampliamente empleado en el sector automovilístico debido a su robustez, inmunidad al ruido gracias a su señal diferencial, y a sus mecanismos integrados de detección de errores. Además, el protocolo incluye un sistema de arbitraje que resuelve las colisiones en el bus, priorizando los mensajes con identificadores (ID) más bajos. Su funcionamiento y estructura han sido descritos en detalle por Corrigan (2016) [6], cuya documentación ha servido de base para este apartado.

La estructura de un mensaje CAN se muestra en la Figura 5. Los campos más relevantes en el contexto de este proyecto son los siguientes: 11-bit identifier o CAN ID. Diferencia el mensaje y le asigna prioridad ante colisión con mensajes con mayor identificador. DLC. Contiene el número de bytes de información que contiene el mensaje Bytes data. Son los bytes de información que variarán según el mensaje, es el propio contenido que vamos a comunicar a los demás módulos.

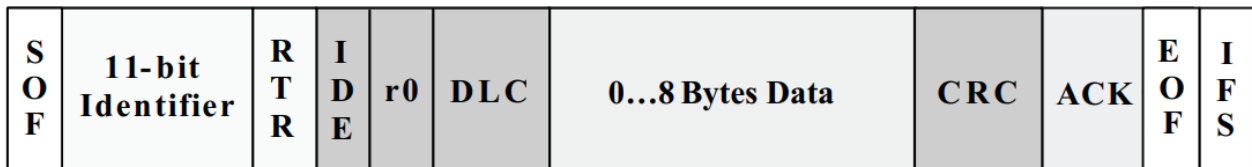


Figura 5. Estructura de un mensaje can. Extraído de Corrigan (2016) [6]

El resto de campos del protocolo no se gestionan directamente desde el microcontrolador, ya que son administrados automáticamente por el circuito integrado MCP2515T, mediante el uso de su correspondiente librería.

La Figura 6 representa la topología de una red CAN. En este tipo de configuración, todos

los módulos se conectan en paralelo a las líneas CANH y CANL, lo que permite que reciban la señal diferencial del bus. Para garantizar la integridad de la señal, se colocan resistencias de terminación de $120\ \Omega$ en cada extremo del bus, tal como establece la especificación del protocolo.

Esta estructura general es la que se adopta conceptualmente en la red de comunicaciones de la motocicleta, cuyo esquema puede verse en la Figura 4. Aunque dicho esquema no muestra de forma explícita la topología física del bus CAN, las conexiones se realizan siguiendo la configuración representada en la Figura 6.

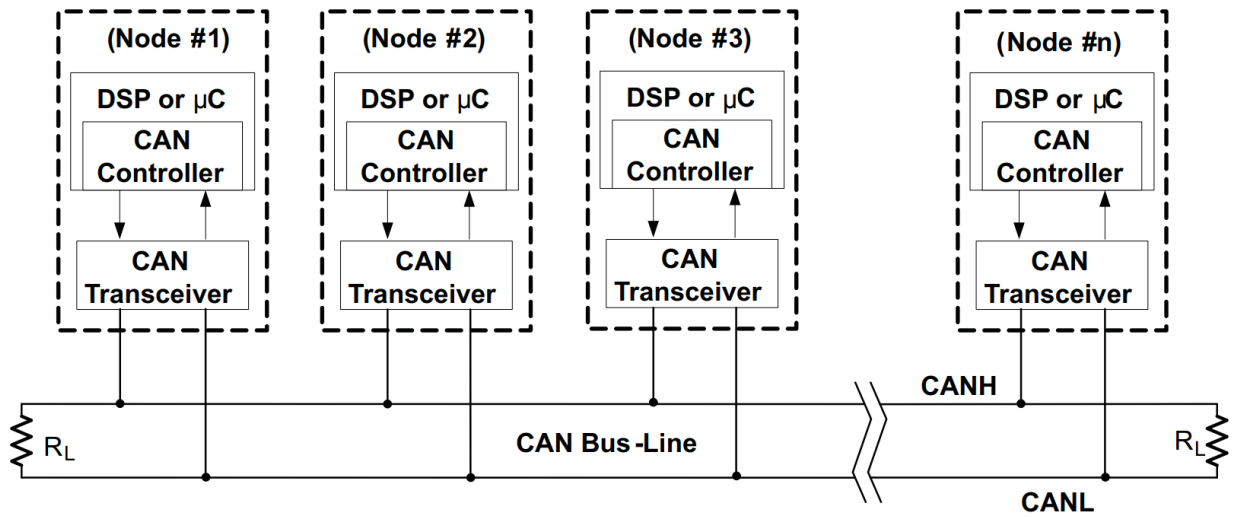


Figura 6. Estructura de una red can. Extraído de Corrigan (2016) [6]

3.2.3. Hardware BMS

El sistema de gestión está formado por dos placas electrónicas (PCB) idénticas, a las que se denominará hardware BMS, que integran un microcontrolador ESP32 comunicado mediante UART con un chip bq76PL455A-Q1. Cada una de estas placas está conectada a un subconjunto del paquete de baterías, sobre el cual realiza la monitorización de tensiones y temperaturas, así como el balanceo pasivo de celdas cuando es necesario. Esta arquitectura permite ampliar la capacidad del sistema manteniendo una estructura modular.

Ambos microcontroladores (Figura 7) adquieren información de sus respectivas celdas y se comunican entre sí mediante el bus CAN para compartir los datos y disponer de una visión completa del estado del paquete. También transmiten esta información al *display* del

vehículo, lo que permite al piloto visualizar en tiempo real los valores clave del sistema. Tras el intercambio de datos, cada ESP32 toma de forma autónoma la decisión de balancear su grupo de celdas, en función de las condiciones globales.

Una de las placas se configura como nodo maestro. Aunque ambas son idénticas a nivel de hardware, su comportamiento se diferencia únicamente por el software programado en cada una. El nodo maestro, además asume el control del proceso de carga mediante la comunicación con el cargador Zivan NG3, así como la lectura ampliada de la botonera del piloto y el controlador del motor. De este modo, coordina el funcionamiento general del sistema, incluyendo la gestión de la carga y la respuesta ante condiciones anómalas.

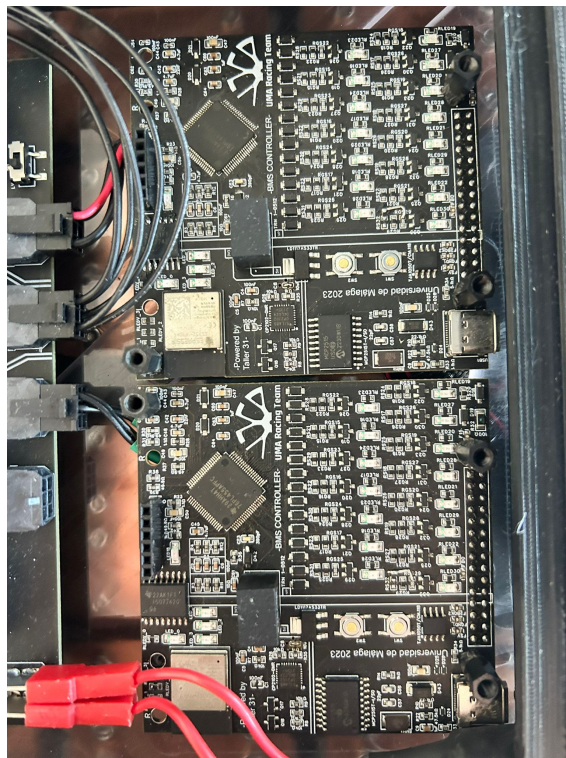


Figura 7. Fotografía de los hardware BMS.

3.2.4. Cargador programable

El sistema de carga (Figura 8) de la batería está gestionado por un cargador industrial de la marca Zivan, modelo NG3, configurado para funcionar con control externo a través del protocolo de comunicación CAN. Este cargador permite ajustar parámetros de tensión máxima y corriente máxima. Además, se puede controlar en tiempo real la corriente de salida,

lo que hace posible la automatización del proceso de carga.

El BMS maestro se comunica directamente con el cargador mediante mensajes CAN, desde los cuales controla la intensidad de carga, establece un límite de tensión objetivo y, en caso necesario, detiene el proceso. Esta interacción es esencial para implementar un perfil de carga de tipo CC–CV, y para garantizar que la batería se carga de forma segura y controlada. Además, el cargador está diseñado para dejar de suministrar corriente si detecta la pérdida de comunicación con el BMS, lo que añade una capa de protección frente a fallos de sistema.



Figura 8. Fotografía del cargador programable.

3.2.5. Pantalla

El vehículo incorpora un *display* de competición Aim MXm (Figura 9), conectado al bus CAN principal. Su función dentro del sistema de gestión eficiente de baterías es exclusivamente informativa, no interviene en la lógica ni en el control del sistema, ni envía mensajes CAN. Esta pantalla permite visualizar en tiempo real los datos más relevantes transmitidos por el bus CAN, como tensiones de celdas o temperaturas. Su configuración es flexible y permite seleccionar fácilmente qué mensajes CAN mostrar, facilitando así la supervisión del estado de la batería y otros parámetros críticos durante la carrera o el proceso de carga.



Figura 9. Fotografía del *display* AimMXm.

3.2.6. Controlador

El controlador utilizado en la motocicleta es un Sevcon Gen4 (Figura 10), encargado de gestionar la entrega de potencia al motor en función de las órdenes del piloto. Aunque su función principal no está directamente relacionada con el sistema de gestión de baterías, sí tiene un papel clave en el control del proceso de carga y en la supervisión del estado energético del vehículo.

El BMS maestro recibe a través del bus CAN la tensión de batería que el Sevcon detecta en su contactor. Esta tensión actúa como referencia para el proceso de carga, ya que el sistema limita la tensión objetivo en función del voltaje leído por el Sevcon. Aunque la batería podría alcanzar tensiones de unos 130V por el número de celdas en serie que la constituye, el controlador establece una restricción crítica: la tensión que detecta no puede superar su umbral de funcionamiento, en torno a los 120V. Por ello, el BMS ajusta la carga y la interrumpe automáticamente cuando la tensión reportada por el Sevcon alcanza el valor objetivo, actuando como mecanismo de seguridad.



Figura 10. Controlador Sevcon gen4. Fuente: [7].

3.2.7. Botonera

La botonera (Figura 11) es una PCB diseñada por el equipo UMA Racing Team que actúa como interfaz física entre el sistema y el usuario. Está conectada al bus CAN y envía distintos mensajes CAN en función de las pulsaciones que recibe. Aunque su diseño es sencillo, permite realizar acciones importantes en la lógica del sistema de gestión eficiente de baterías. Entre sus funcionalidades se encuentra la posibilidad de activar o desactivar manualmente el balanceo de celdas, así como modificar el valor de corriente que inyecta el cargador durante la carga.



Figura 11. Fotografía de la botonera.

Capítulo 4

Hardware BMS

4.1. Arquitectura general del hardware

El sistema de gestión de baterías desarrollado en este proyecto se implementa sobre una PCB diseñada por el equipo UMA Racing Team (Figura 1). Esta PCB se basa en la placa de evaluación de Texas Instruments para el chip bq76PL455A-Q1, descrita en la documentación oficial del fabricante [8], aunque incorpora mejoras orientadas a su integración en la motocicleta de competición como mayor conectividad y comunicación CAN.

El sistema emplea dos unidades idénticas de este hardware, configuradas de forma modular, ya que el chip bq76PL455A-Q1 está diseñado para monitorizar y equilibrar hasta un máximo de 16 celdas. Esta arquitectura permite escalar el sistema fácilmente: en el caso de esta motocicleta, que dispone de 30 celdas, se emplean dos PCBs que gestionan 15 celdas cada una. Cada PCB integra un microcontrolador ESP32, encargado de ejecutar el software del BMS.

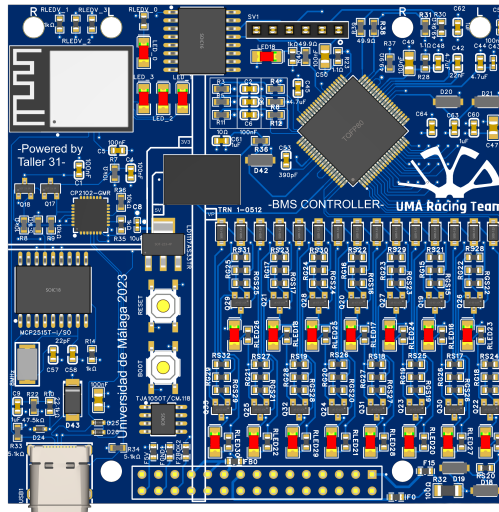


Figura 1. Modelo 3D del hardware del BMS.

4.2. ESP32

4.2.1. Integración en el sistema

El ESP32 actúa como unidad central de decisión dentro de cada hardware de BMS (Figura 2). Tras adquirir datos del circuito integrado bq76PL455A-Q1, es responsable de analizarlos y activar las funciones necesarias, como el balanceo de celdas o el envío de información a otros módulos. Esta lógica de actuación convierte a los ESP32 en los controladores del sistema de gestión de baterías.

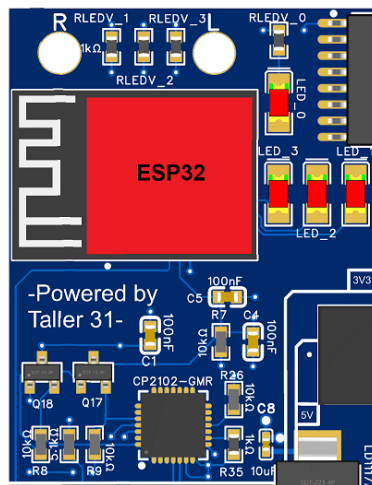


Figura 2. Periferia del ESP32 en el hardware del BMS.

En el sistema implementado un ESP32 actúa como esclavo, dedicado exclusivamente a la monitorización y balanceo de su conjunto de celdas y posterior envío de información al otro BMS y al *display*. Otro de los ESP32 adopta un rol de nodo maestro, asumiendo funciones adicionales de comunicación con el controlador y el cargador. Esta distribución funcional se realiza únicamente a nivel de software, ya que ambas placas son físicamente idénticas.

4.2.2. Comunicación con el sistema

La comunicación con el chip bq76PL455A-Q1 se realiza mediante UART diferencial aislada, utilizando un circuito aislador ISO7742QDWQ1 (Figura 5). La velocidad de comunicación es 250000 baudios. Esta interfaz permite al ESP32 adquirir de forma segura y en tiempo real las tensiones de cada celda, temperaturas del sistema y estado del balanceo, constituyendo el punto de partida de toda la lógica de decisión local que ejecuta el microcontrolador.



Figura 3. Modelo 3D de MCP2515T en el hardware del BMS.

Además de su interacción directa con el chip bq, el ESP32 gestiona la comunicación con el resto de módulos de la motocicleta a través del bus CAN. La velocidad de comunicación CAN de todos los módulos es de 500Kbits. Esta interfaz permite transmitir información al *display*, coordinarse con las demás PCBs del BMS y establecer el control sobre elementos como el cargador y la botonera. Aunque el ESP32 no genera directamente señales en protocolo CAN, emplea una interfaz SPI para enviar los mensajes a un circuito integrado MCP2515T (Figura 3), que los convierte a tramas CAN digitales. A continuación, un transceptor TJA1050T adapta estas señales al nivel físico del bus, garantizando una comunicación robusta y fiable con el sistema eléctrico de la motocicleta.

4.3. Monitor de celdas bq76PL455A-Q1

El bq76PL455A-Q1 (Figura 4) es un circuito integrado de Texas Instruments diseñado para la monitorización, protección y balanceo de hasta 16 celdas en serie en sistemas de baterías utilizados en vehículos eléctricos, como se detalla en la hoja de datos oficial [9].

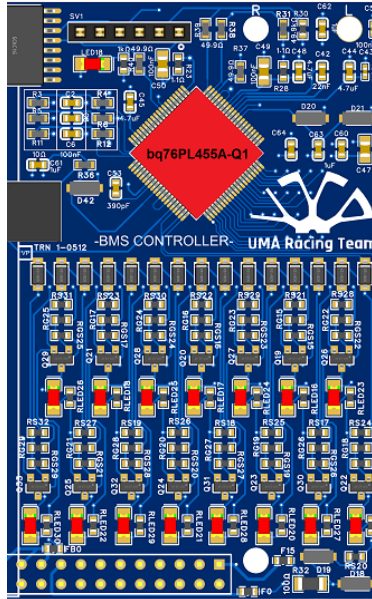


Figura 4. Periferia del bq76PL455A-Q1 en el hardware del BMS.

4.3.1. Características técnicas

Incorpora un convertor analógico-digital de 14 bits que permite medir los voltajes individuales de las celdas con un rango de medición comprendido entre 1 V y 4.95 V, una precisión de ± 0.75 mV a 25 °C. El dispositivo puede realizar la conversión de los 16 canales de celda en 2.4 ms en condiciones nominales, lo que permite obtener lecturas rápidas y fiables del estado de la batería. Además de las entradas dedicadas a celdas, dispone de ocho entradas analógicas auxiliares para la lectura de sensores de temperatura u otras señales, con un rango de entrada de 0 V a 5 V y una precisión típica de ± 0.1 mV a temperatura ambiente. Estas entradas son útiles para implementar sistemas de supervisión térmica distribuida en el sistema de baterías. El chip incluye salidas digitales dedicadas para el control de balanceo pasivo por cada celda, accesibles mediante las líneas EQ1 a EQ16, con activación programable. Estas salidas

EQ controlan el balanceo sobre resistencias de 75Ω que proporcionan unas corrientes de balanceo de 56mA, estas corrientes son bastante reducidas para las dimensiones del paquete de baterías empleado. Desde el punto de vista eléctrico, el bq76PL455A-Q1 puede soportar hasta 88 V en su entrada superior TOP, lo que lo hace adecuado para sistemas de baterías de vehículos eléctricos. Además, presenta una inmunidad a descargas electrostáticas de hasta 2 kV bajo el modelo HBM.

4.3.2. Comunicación con el sistema

El protocolo de comunicación del bq76PL455A requiere de UART diferencial aislado (Figura 5). La UART diferencial garantiza inmunidad al ruido en entornos electromagnéticamente hostiles, como el compartimento de baterías de un vehículo eléctrico. Esta topología permite una comunicación punto a punto fiable entre el ESP32 y el bq76PL455A-Q1. El protocolo del chip bq76PL455A-Q1 es específico y viene descrito en sus hojas de datos. Los mensajes empleados son descritos en las funciones correspondientes.

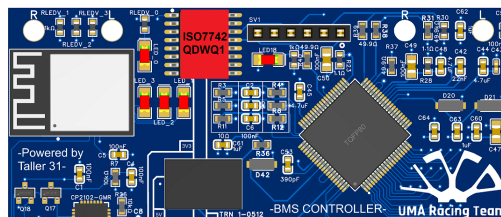


Figura 5. Modelo 3D de aislador de UART en el hardware del BMS.

4.3.3. Integración en el sistema

El circuito integrado bq76PL455A-Q1 se encarga de la monitorización dentro del sistema de gestión de baterías de la motocicleta. Su principal función es medir en tiempo real las tensiones de cada una de las celdas conectadas en serie, así como las temperaturas de diversos puntos del paquete de baterías. Además, cumple con la tarea de balancear pasivamente los grupos de celdas para garantizar la seguridad y eficiencia del conjunto de paquetes que conforman la batería.

El dispositivo se conecta directamente a los terminales de cada celda a través de las entradas VSENSE0 a VSENSE16, permitiendo así obtener el voltaje diferencial entre celdas

consecutivas. Adicionalmente, se emplean tres de las ocho entradas auxiliares AUX0–AUX7 para adquirir las señales de los sensores de temperatura ubicados en la caja de la batería.

El chip se comunica con el microcontrolador ESP32 mediante UART con una configuración esclavo-maestro. Esta comunicación permite no solo la lectura de valores, sino también el envío de comandos para habilitar o deshabilitar el balanceo de celdas. Durante la inicialización del microcontrolador, el ESP32 transmite una configuración al bq76PL455A-Q1. A partir de ese momento, el ESP32 puede controlarlo en cualquier instante mediante comandos específicos. Las respuestas son devueltas por el chip a través del mismo canal de comunicación, permitiendo así una monitorización flexible y adaptada a las necesidades del sistema.

4.3.4. Funciones del bq76PL455A-Q1

El ESP32 de cada hardware del BMS emplea una librería para gestionar y transmitir la información obtenida a través del circuito integrado de monitorización y protección bq76PL455A-Q1. Todas las funciones descritas en este apartado se han desarrollado a partir de los mensajes y estructuras de datos definidos en la documentación de software proporcionada por Texas Instruments [10]. Las funciones más relevantes se exponen a continuación.

4.3.4.1. InitializeBMS()

Esta función se encarga de realizar la configuración inicial del circuito integrado bq76PL455A-Q1. Aunque el dispositivo permite distintas configuraciones según las necesidades del sistema, en este proyecto se ha optado por una configuración básica que, en gran medida, coincide con los valores por defecto del propio dispositivo. Requiere del envío de los siguientes mensajes UART:

`0x F1 13 00`

0xF1 denota un mensaje broadcast de un solo byte.

0x13 es el registro CBCONFIG, o Balance Configuration.

0x08 configura el bq76PL455A-Q1 para que el ESP32 tenga total control sobre el balanceo de celdas y no necesite mensajes periódicos para balanceo.

Bit	Descripcion
[7:4]	Timer para que el balanceo se apague solo si no tiene supervisión. 0 = deshabilitado.
[3]	Restringe el balanceo si detecta un error. 0 = restringe, 1 = no restringe.
[2:0]	reservado, siempre a 0.

Tabla 1. Registro CBCONFIG.

0x 94 00 03 00 byte0 byte1 byte2 byte3

0x94 denota un mensaje sin respuesta de cuatro bytes.

0x00 es la dirección del dispositivo al que se comunica el mensaje.

0x03 es el registro CHANNELS.

byte0, byte1, byte2 y byte3 son los bytes de datos, estos varían según el paquete de baterías que queramos manejar puesto que el código es flexible. Típicamente con 15 celdas y 3 sensores, se envían los bytes de datos 7F 07 00 00.

Bit	Descripcion
[31:16]	Una máscara para el bq76PL455A-Q1 que hace que las celdas sean sensibles a comandos. Cada bit corresponde a una celda. 1 = esta celda es sensible, 0 = este canal no es sensible.
[15:8]	Una máscara para el bq76PL455A-Q1 que hace que los canales auxiliares sean sensibles a comandos. Cada bit corresponde a un canal auxiliar. 1 = este canal es sensible, 0 = este canal no es sensible.
[7]	temperatura interna digital, para testeo.
[6]	temperatura interna analogica, para testeo.
[5]	fuentes de alimentación interna, para testeo
[4]	reservado, siempre a 0.
[3]	reservado, siempre a 0.
[2]	referencia de tensión para autotesteo.
[1]	suma de tension de las celdas, para autotesteo.
[0]	monitor de voltajes, para autotesteo.

Tabla 2. Registro CHANNELS.

0x 91 00 0D 00 bytecells

0x91 denota un mensaje sin respuesta de un byte.

0x00 es la dirección del dispositivo al que se comunica el mensaje.

0x0D es el registro NCHAN.

bytecells es el byte de datos, este varía según el paquete de baterías que queramos manejar puesto que el código es flexible. Típicamente con 15 celdas, se envía 0x0F.

Bit	Descripcion
[7:5]	reservado, siempre a 0.
[4:0]	Número de canales VSENSE que se usan. Los inutilizados empiezan desde el canal 16.

Tabla 3. Registro NCHAN.

4.3.4.2. RequestSample()

Esta función envía el comando al chip bq76PL455A-Q1 para que guarde las muestras configuradas en un registro específico para esto .

0x F1 02 00

0xF1 denota un mensaje broadcast sin respuesta de un byte.

0x02 es el registro CMD.

0x00 Envía el comando SYNC SAMPLE para que tome una muestra de forma síncrona en todos los canales previamente habilitados.

4.3.4.3. ReadData()

Esta función envía el comando al chip bq76PL455A-Q1 para que responda con las muestras previamente almacenadas.

0x 81 00 02 20

0x81 denota un mensaje con respuesta.

0x00 es la dirección del dispositivo al que se comunica el mensaje.

0x02 es el registro CMD.

0x20 Envía el comando READ SAMPLED VALUES para que el bq76PL455A-Q1 inicie una respuesta con las muestras previamente almacenadas.

4.3.4.4. **balanceCells()**

Esta función envía al bq76PL455A-Q1 que celdas específicas se deben balancear.

0x 92 00 14 HighCells LowCells

0x92 denota un mensaje sin respuesta de dos bytes.

0x00 es la dirección del dispositivo al que se comunica el mensaje.

0x14 es el registro CCBENBL.

HighCells y LowCells controla las celdas específicas en las que se desea activar el balanceo pasivo. Este mensaje es variable según se determine en el algoritmo de balanceo.

Bit	Descripcion
[15:0]	0 = balanceo desactivo en este canal. 1 = balanceo activado en este canal.

Tabla 4. Registro CCBENBL.

4.3.4.5. **RawDataToProcessData()**

Esta función transforma los bytes de respuesta de ReadData() a una estructura de datos con un formato más manejable en la programación.

4.3.4.6. **CommunicationCANDisplay()**

Esta función envía los valores individuales de tensión por cada celda y temperatura por cada sensor por CAN. El formato de este mensaje es compatible con la programación del *display* AimMxM para que se muestren por pantalla los valores.

4.3.4.7. **getInfoData()**

Esta función obtiene las tensiones máximas, mínimas y totales de RawDataToProcessData() y las almacena en una estructura de datos.

4.3.4.8. **sendInfoData()**

Esta función envía la estructura de datos empleada en getInfoData() por CAN al resto de módulos de la motocicleta.

4.3.4.9. `readOtherBMSData()`

Esta función lee la estructura de datos que envía las esp32 de otros hardware BMS mediante `sendInfoData()` y guarda su contenido en la memoria, de esta forma, de esta forma puede conocer las tensiones máximas, mínimas y totales, y las temperaturas de todas las demás PCBs BMS.

4.3.4.10. `autobalance()`

Esta función implementa el algoritmo descrito en el apartado Algoritmo de balanceo. A partir de la lectura de tensiones obtenidas de los ESP32 y comunicándose entre ellos, se determina la tensión mínima entre todas las celdas del sistema. A continuación, se recorre cada celda y se compara su tensión con respecto al valor mínimo. Si la diferencia supera un umbral definido, la celda se marca para ser balanceada. Una vez identificadas todas las celdas que requieren balanceo, se llama a la función `balanceCells()` para activar el proceso de balanceo en los canales correspondientes.

4.4. Algoritmo de balanceo

El algoritmo de balanceo empleado en este proyecto está basado en voltaje. La referencia se establece a partir de la celda con menor voltaje, y se activa el balanceo en todas aquellas celdas cuya tensión supere dicho valor más un umbral predefinido. El proceso continúa hasta que la diferencia entre la celda con mayor tensión y la celda con menor tensión es inferior a un error máximo programado por el usuario. En ese momento, se activa una señal que indica que el balanceo ha finalizado.

El algoritmo empleado en este proyecto se caracteriza por su sencillez, robustez y velocidad de ejecución. Existen alternativas más complejas que pueden ofrecer mejoras en términos de capacidad efectiva del paquete de celdas, grado de equilibrio alcanzado y energía disipada durante el proceso de ecualización. No obstante, estos algoritmos son más lentos y complejos de implementar. Tal y como señala De la Vega (2025) [11], las diferencias en los parámetros optimizados por los distintos algoritmos de balanceo pasivo suelen tener un impacto muy reducido sobre el comportamiento global del sistema. En la mayoría de los casos, las mejoras

se limitan a incrementos porcentuales mínimos o a variaciones de tensión en el rango de los microvoltios, lo que resulta despreciable en aplicaciones prácticas como la que se aborda en este proyecto.

Los ensayos realizados reflejan que el algoritmo y el hardware desarrollados en este proyecto consiguen una diferencia de tan solo 2 mV entre la celda con mayor y menor tensión, un resultado que destaca por su precisión. Un BMS comercial como puede ser Orion BMS, establece en su manual técnico [12] que el proceso de balanceo debe finalizar cuando la diferencia entre celdas desciende por debajo de 10 mV. En el caso de los sistemas de fabricantes de automoción, en vehículos eléctricos como el Nissan Leaf o el Chevrolet Bolt registran diferencias residuales tras el balanceo situadas entre 8 mV y 12 mV. Por tanto, los errores obtenidos en este trabajo no solo se sitúan por debajo de lo recomendado por sistemas comerciales reconocidos, sino también por debajo de los valores estimados en vehículos de producción masiva, lo que confirma la precisión alcanzada por el diseño y señala que una mejora adicional solo sería posible mediante cambios en el hardware de medida.

4.5. Mejoras implementadas sobre el diseño base

El hardware del BMS utilizado en este proyecto fue desarrollado previamente por el equipo UMA Racing Team, aunque nunca llegó a utilizarse para ejecutar funciones de balanceo de celdas. Al implementar dicha funcionalidad, se identificaron dos problemas clave que requerían intervención a nivel de hardware.

El primero de ellos estaba relacionado con un error de diseño en la electrónica de balanceo correspondiente a una de las celdas. Este hardware toma como base la placa de evaluación oficial de Texas Instruments para el chip bq76PL455A-Q1, cuyo esquemático se muestra en la Figura 6. Al comparar con el diseño del UMA Racing Team, representado en la Figura 7, se observa que la resistencia RGS29, equivalente a la R137 en la placa de evaluación de Texas Instruments, está mal ubicada. Esta disposición impide que se efectúe el balanceo sobre la celda correspondiente al pin EQ1 del chip. Este error fue corregido de forma inmediata.

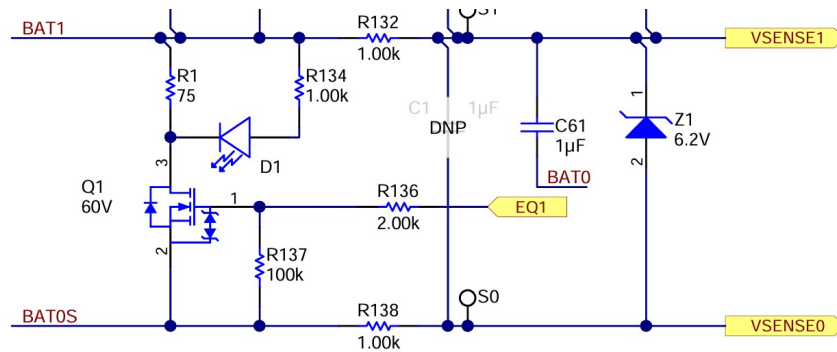


Figura 6. Electrónica de la placa de desarrollo del bq76PL455A-Q1.

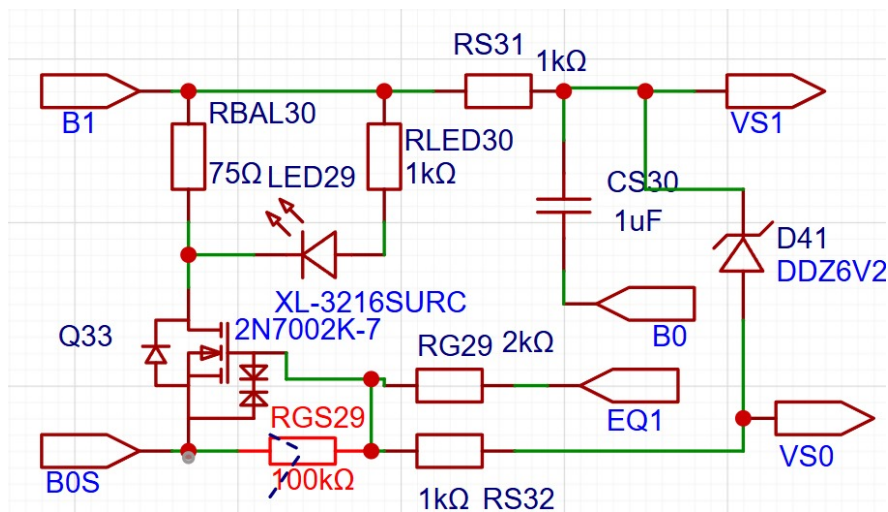


Figura 7. Electrónica del hardware del BMS de UMA Racing Team.

El segundo problema afectaba directamente al rendimiento del balanceo. Originalmente, las resistencias utilizadas en el balanceo eran de $75\ \Omega$, lo que limitaba la corriente de balanceo a aproximadamente $56\ \text{mA}$. En consecuencia, el balanceo podía prolongarse durante decenas de horas para corregir desajustes de apenas $10\ \text{mV}$ entre celdas. Para solventarlo, se analizaron posibles sustituciones de resistencias SMD que mantuvieran la disipación de potencia dentro del umbral seguro de $1\ \text{W}$. Finalmente, se optó por resistencias de $16\ \Omega$, lo que permite alcanzar corrientes de balanceo de hasta $250\ \text{mA}$, multiplicando por cinco las corrientes de balanceo sin comprometer la integridad térmica del sistema. Este cambio supone una mejora significativa en la eficiencia del balanceo pasivo implementado y será aplicada a la próxima generación de BMS de UMA Racing Team.

Capítulo 5

Cargador programable

El cargador programable es un Zivan NG3-G7MICB. Este es un cargador monofásico diseñado para aplicaciones que requieren la gestión de carga de baterías con control mediante comunicación digital como por ejemplo CAN.

5.1. Características técnicas

Como se detalla en la hoja de datos del fabricante [13], el cargador Zivan NG3 ofrece una potencia máxima de salida de 2,9kW. En el contexto de este proyecto, la batería se carga generalmente a una tensión cercana a 120V, lo que permite alcanzar corrientes de carga de hasta 24A. Opera con una tensión de alimentación de 230 VAC $\pm 10\%$ o 115 VAC $\pm 10\%$, y una frecuencia de entrada de 50 a 60 Hz, lo que lo hace compatible con las redes eléctricas tanto en Europa como en América. Su rendimiento superior al 85% garantiza un bajo consumo energético y una alta eficiencia de conversión.

El equipo está diseñado para operar en un amplio rango de temperaturas, desde -20 °C hasta +50 °C, lo que permite su funcionamiento en diversas condiciones ambientales. En cuanto a seguridad, incorpora protecciones contra cortocircuito en la salida y contra inversión de polaridad, ambas mediante fusibles, asegurando la integridad del sistema y de la batería conectada. El sistema cuenta con ventilación forzada, para aumentar la disipación térmica bajo condiciones de carga intensiva.

La curva de carga es programable, permitiendo su adaptación a distintos tipos de baterías

y perfiles de carga. Asimismo, ofrece una desviación máxima de $\pm 0,5\%$ en la tensión de salida. Además, el cargador está equipado con una interfaz CAN BUS aislada, la cual será empleada en este trabajo para comunicar el cargador con el BMS.

5.2. Comunicación con el sistema

El cargador tiene diferentes métodos de programación y de comunicación. En este caso se empleará la comunicación por CAN para programar y controlar el cargador.

El cargador tiene distintas opciones de Firmware. Inicialmente estaba programada con el Firmware CB [14] que permitía controlar el cargador con mensajes CAN (Figura 1), sin embargo, el uso de este protocolo no viene detallado y está más orientado a usar una aplicación que ofrece la misma empresa.

Por este motivo se realizó el cambio al firmware RE [15], que está pensado para el control del cargador a través de unidades externas al proceso de carga, como un BMS. El firmware RE permite al BMS controlar las intensidades y los voltajes máximos que opera el cargador a través de mensajes CAN reservados.

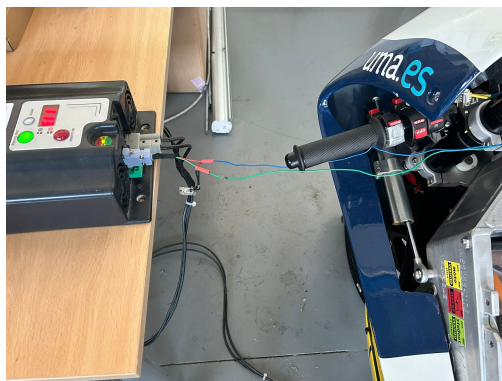


Figura 1. Conexión CAN del cargador con la motocicleta.

5.3. Integración en el sistema

La finalidad del cargador es suministrar energía al paquete de baterías hasta alcanzar el punto de carga objetivo establecido por el BMS. Para ello es necesario establecer dos conexiones. Una conexión para transferir potencia con la batería (Figura 2) y una conexión

para realizar la comunicación con el bus CAN de la motocicleta (Figura 1). Esta última permite al sistema de gestión de baterías interactuar con el cargador de forma controlada y segura.



Figura 2. Conexión de la potencia del cargador con la motocicleta.

La detección de la presencia del cargador se realiza automáticamente por parte del BMS mediante la lectura de mensajes en el bus CAN. Una vez establecida la comunicación, el BMS asume el control del proceso de carga mediante el envío de comandos específicos. Algunos de estos comandos, como la configuración de los límites de tensión y corriente, se transmiten únicamente al inicio de la comunicación. En cambio, otros parámetros operativos, como la consigna de corriente en tiempo real, requieren un envío periódico para mantener activo el suministro.

El funcionamiento del cargador se encuentra subordinado al control del BMS. Por razones de seguridad, el firmware RE del Zivan NG3 requiere que el BMS le envíe periódicamente una consigna de corriente. Si transcurren más de cuatro segundos sin recibir dicha instrucción, el cargador interrumpe automáticamente el suministro de corriente hacia la batería, previniendo posibles condiciones de operación insegura en caso de pérdida de comunicación.

5.3.1. Funciones del Zivan NG3-G7MICB

El ESP32 maestro de los hardware BMS emplea una librería para comunicarse y controlar por CAN el cargador Zivan NG3-G7MICB. Esta librería está diseñada para controlar

el cargador como una máquina de estados. Los posibles estados son: desconectado, inicio, configuración, precarga, carga_CC, carga_CV y carga_fin. Las funciones más relevantes se exponen a continuación.

5.3.1.1. procesarCan()

Esta función se encarga de identificar todos los mensajes que provienen del cargador programable, actuando sobre señales de control y actualizando variables internas en función de los datos recibidos. Además, incorpora la lógica de detección automática de la presencia del cargador cuando detecta el mensaje CAN con ID 0x380. Este mensaje activa la señal cargadorConectado que indica su conexión. Si no se recibe ningún mensaje durante un intervalo de tiempo predefinido, la función desactiva cargadorConectado, interpretando que el cargador ha sido desconectado.

Byte	Descripción
[0:1]	reservado, siempre a 0.
[2:3]	Corriente actual del cargador.
[4:5]	Tensión actual del cargador.
[6:7]	Amperios hora cargados en esta sesión.

Tabla 1. Mensaje CAN 0x380.

5.3.1.2. comprobar_conexion()

Esta función comprueba que procesarCan() ha detectado una conexión y cambia el estado del cargador a inicio.

5.3.1.3. enviar_configuracion_inicial()

Envía el mensaje broadcast CAN 0x776 al cargador configurando corriente, tensión máxima y límite inferior de tensión para detectar una batería por parte del cargador.

Byte	Descripción
[0:1]	Corriente máxima que se quiere programar.
[2:3]	Tensión máxima que se quiere programar.
[4:5]	No se usan.
[6:7]	Umbral de tensión para reconocer una batería que se quiere programar.

Tabla 2. Mensaje CAN 0x776.

5.3.1.4. comprobar_configuracion()

Esta función comprueba que procesarCan() ha detectado que el cargador ha cambiado correctamente su configuración en el mensaje 0x180 y cambia el estado del cargador a precarga o directamente a carga_CC según requiera el usuario.

Byte	Descripción
[0:1]	Corriente máxima programada.
[2:3]	Tensión máxima programada.
[4]	Fin de precarga. 1 = Precarga finalizada.
[5]	'O'.
[6]	'K'.
[7]	Nodo del cargador.

Tabla 3. Mensaje CAN 0x180.

5.3.1.5. enviar_precarga()

Envía el mensaje CAN 0x600 al cargador configurando la tensión de precarga.

Byte	Descripción
[0]	'22'
[1]	'60'
[2]	'60'
[3]	'01'
[4:5]	Tensión de precarga.
[6]	'0'
[7]	'0'

Tabla 4. Mensaje CAN 0x600 inicio precarga.

5.3.1.6. comprobar_precarga()

Esta función comprueba que procesarCan() ha detectado el mensaje 0x180 con byte 4 a '1'. Esto significa que el cargador ha finalizado la precarga y cambia el estado del cargador a carga_CC".

5.3.1.7. finalizar_precarga()

Envía el mensaje CAN 0x600 al cargador con los bytes necesarios para terminar la precarga e iniciar la carga normal.

Byte	Descripción
[0]	'22'
[1]	'60'
[2]	'60'
[3]	'01'
[4]	'0'
[5]	'0'
[6]	'0'
[7]	'0'

Tabla 5. Mensaje CAN 0x600 fin precarga.

5.3.1.8. `enviar_porcentaje_corriente()`

Envía el mensaje CAN 0x6C1 al cargador controlando el porcentaje de corriente, respecto a la corriente máxima configurada, que debe inyectar el cargador a la batería.

Byte	Descripción
[0:5]	'0'.
[6:7]	Consigna de porcentaje de corriente máxima para el cargador.

Tabla 6. Mensaje CAN 0x6C1.

5.3.1.9. `ejecutar_carga_CC()`

Envía cada dos segundos el mensaje `enviar_porcentaje_corriente()`. Si el cargador no recibe este mensaje a los cuatro segundos, se apaga automáticamente por seguridad. Esta función a su vez detecta que la tensión del sevcon está aproximándose a su tensión objetivo y cambia el estado del cargador a `carga_aproximacion_objetivo`.

5.3.1.10. `calcular_media_vector()`

Calcula la media de un vector de tensiones, en este caso, se usa para calcular la media de tensión de una ventana de muestras de tensiones del controlador Sevcon. Esta media se usa como filtro paso bajo digital, debido a que las muestras que envía el Sevcon puede variar en decimas a causa del ruido en la toma de datos.

5.3.1.11. `calcular_porcentaje_corriente_CV()`

Controla que porcentaje de intensidad máxima debe circular del cargador a la batería cuando se aproxima a la tensión objetivo. Es un controlador proporcional al que posteriormente se le añade un 20% de intensidad. La actuación satura por encima de cien y por debajo de cero.

5.3.1.12. `ejecutar_carga_CV()`

Envía cada dos segundos el mensaje `enviar_porcentaje_corriente()` con el valor `calcular_porcentaje_corriente_CV()`. Si el cargador no recibe este mensaje a los cuatro segundos, se apaga automáticamente por seguridad. Esta función a su vez detecta que la tensión del

sevcon llega a su objetivo y cambia el estado del cargador carga_fin.

5.3.1.13. detener_cargador()

Esta función envía una intensidad de cero con el mensaje enviar_porcentaje_corriente() y se bloquea la maquina de estados. Espera a que el usuario envíe de nuevo una señal de nueva carga o se desconecte el cargador para desbloquear el cargador.

Capítulo 6

Pantalla

6.1. Características técnicas

AimMXm es un dispositivo compacto orientado a aplicaciones de automoción de alto rendimiento, que combina funciones de *display* y adquisición de datos en un único módulo. Según la documentación técnica proporcionada por el fabricante [16], cuenta equipado con una pantalla gráfica de 268 por 128 píxeles, retroiluminada mediante LEDs RGB configurables en siete colores, y permite definir hasta ocho páginas de visualización independientes. Incorpora cinco LEDs RGB programables y dos LEDs adicionales que pueden configurarse como indicadores de alerta (Figura 1).



Figura 1. Modelo de AimMxM. Fuente: AIM [16].

El sistema incluye un módulo GPS con base de datos de circuitos precargada, memoria interna de 4 GB para registro de datos y conectividad Wi-Fi para descarga de datos y configuración. Dispone además de entradas analógicas y digitales, así como salidas de potencia que permiten interactuar directamente con sensores y actuadores externos. Su carcasa, fabricada en nailon reforzado con fibra de vidrio, proporciona una protección frente al polvo y las salpicaduras conforme al grado IP65. Está diseñada para resistir condiciones de vibración, temperatura y humedad típicas en entornos de competición.

6.2. Comunicación con el sistema

Este apartado se ha redactado siguiendo las especificaciones del fabricante descritas en el manual técnico [17]. La pantalla Aim MXm se comunica con el BMS a través del bus CAN de la motocicleta. Su función consiste en mostrar por pantalla información relevante para el usuario. Entre los parámetros mostrados relacionados con el sistema de gestión de baterías se incluyen las tensiones individuales, las temperaturas y otros datos relacionados el sistema.

El Aim MXm solo recibe información. No transmite mensajes CAN en ningún momento, por lo que no participa en la lógica de control del sistema. La configuración necesaria

para interpretar correctamente los mensajes y asociarlos a variables visualizables se realiza mediante el programa RaceStudio 3, proporcionado por el fabricante. Para descargar e interactuar con los datos almacenados en la pantalla, así como para actualizar su configuración y programación, se utiliza la conexión Wi-Fi integrada en el dispositivo.

Para que las variables programadas en la pantalla se mantengan visibles y actualizadas, es necesario que los mensajes CAN correspondientes se transmitan de forma periódica. Aunque el programa RaceStudio 3 no permite configurar explícitamente un tiempo de expiración para cada variable, el sistema interno del Aim MXm. Por este motivo, los mensajes CAN enviados por el sistema BMS deben refrescarse cada pocos segundos para garantizar una visualización continua y fiable de los datos.

6.3. Integración en el sistema

La integración de la pantalla se ha realizado también con el apoyo de la documentación oficial proporcionada por el fabricante [17]. Como ya se ha mencionado, la pantalla Aim MXm actúa exclusivamente como dispositivo receptor y se encarga de proyectar en pantalla la información que recibe para que pueda ser visualizada por el piloto o el equipo técnico. Para que el dispositivo asocie correctamente los mensajes CAN con los parámetros que se desean mostrar, es necesario definir un protocolo de comunicación específico. Esta configuración se realiza mediante el software RaceStudio 3, que presenta ciertas limitaciones en cuanto a flexibilidad, ya que tiene opciones limitadas definidas por el fabricante. No obstante, el proceso de configuración es sencillo y está diseñado específicamente para competición, lo que facilita su integración en la motocicleta.

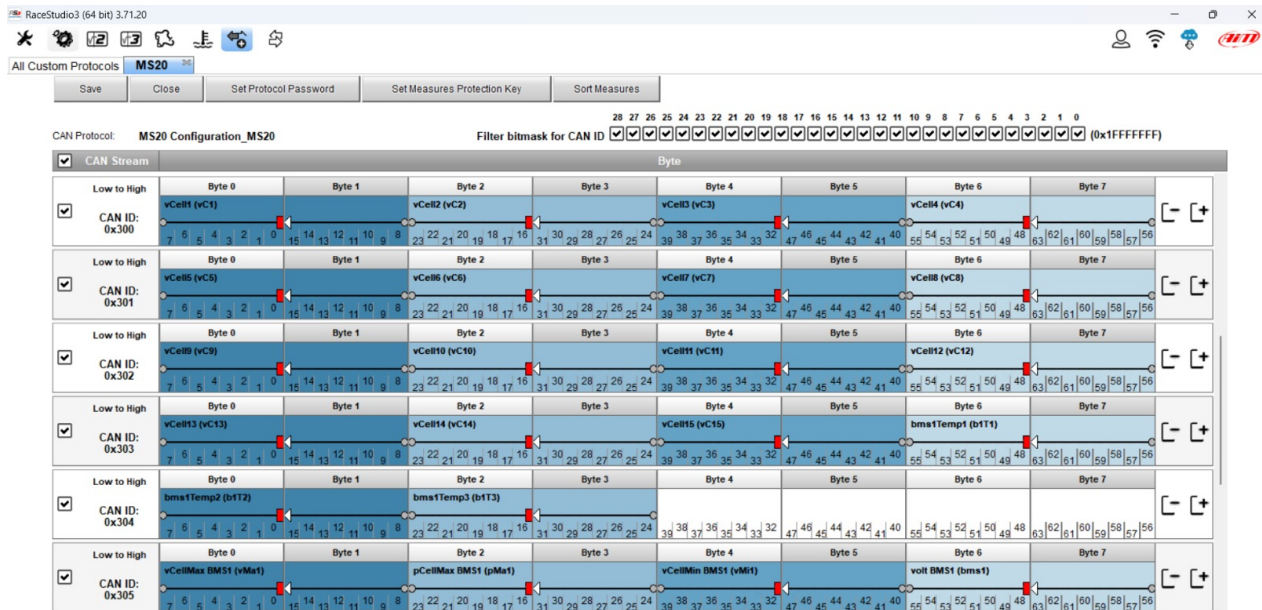


Figura 2. Fragmento del protocolo programado RaceStudio 3.

Como ejemplo, en la Figura 2 se muestra un fragmento del protocolo implementado en este proyecto para la visualización de datos. Los mensajes con identificadores 0x300, 0x301, 0x302, 0x303, 0x304 y 0x305 son enviados por el ESP32 maestro del sistema BMS. En esta configuración, cada identificador CAN se vincula a un conjunto de bytes que el Aim MXm interpreta como variables específicas. La Tabla 1 ilustra cómo se ha programado el dispositivo para asociar los bytes del mensaje con ID 0x300 a diferentes parámetros que posteriormente se representan gráficamente en pantalla.

Byte	Descripción
[0:1]	Tensión de la primera celda.
[2:3]	Tensión de la segunda celda.
[4:5]	Tensión de la tercera celda.
[6:7]	Tensión de la cuarta celda.

Tabla 1. Mensaje CAN 0x300.

Tiene un límite de ocho distintas hojas de visualización que se comparten entre todas las funcionalidades de la motocicleta. La programación de estas hojas de visualización se hace con plantillas que ofrece el programa en las que se relacionan las celdas que se muestran con

las variables que se ha captado en el protocolo programado previamente. Como ejemplo en la Figura 3 se visualiza la hoja que se emplea durante la carrera. Se muestra la velocidad de la motocicleta, la tensión del controlador y la temperatura del motor.

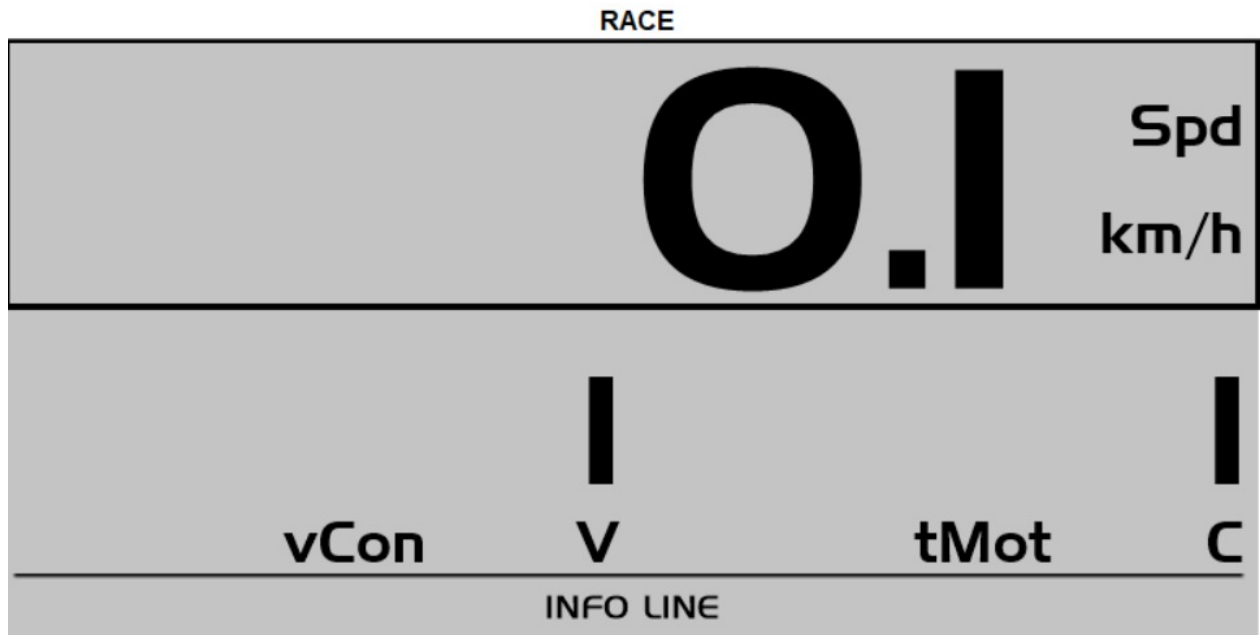


Figura 3. Hoja de visualización de carreras.

El *display* emplea cuatro hojas dedicadas a la supervisión del sistema de gestión de baterías. En las Figuras 4 y 5 se representan las tensiones individuales de todas las celdas del paquete. La Figura 6 muestra las temperaturas de los sensores ubicados en la batería, en el motor y en el controlador.

BMS 1 - CELLS							
0.00		0.00		0.00		0.00	
vC1	V	vC2	V	vC3	V	vC4	V
0.00		0.00		0.00		0.00	
vC5	V	vC6	V	vC7	V	vC8	V
0.00		0.00		0.00		0.00	
vC9	V	vC10	V	vC11	V	vC12	V
0.00		0.00		0.00			--
vC13	V	vC14	V	vC15	V		
INFO LINE							

Figura 4. Hoja de visualización de primeras quince celdas.

BMS 2 - CELLS							
0.00		0.00		0.00		0.00	
vC16	V	vC17	V	vC18	V	vC19	V
0.00		0.00		0.00		0.00	
vC20	V	vC21	V	vC22	V	vC23	V
0.00		0.00		0.00		0.00	
vC24	V	vC25	V	vC26	V	vC27	V
0.00		0.00		0.00			--
vC28	V	vC29	V	vC30	V		
INFO LINE							

Figura 5. Hoja de visualización de últimas quince celdas.

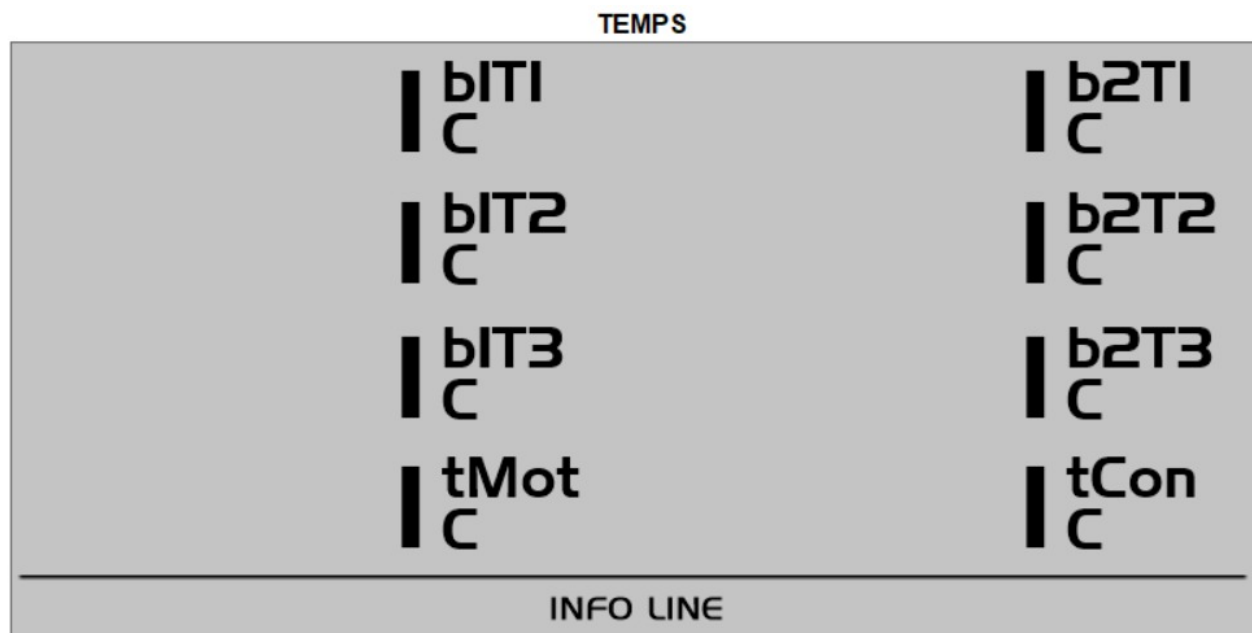


Figura 6. Hoja de visualización de temperaturas.

En la Figura 7 se muestra la hoja de visualización general del sistema BMS. En ella se representa la tensión total de la batería, la temperatura media de los sensores de la batería, el estado de balanceo ON/OFF y el estado del cargador.

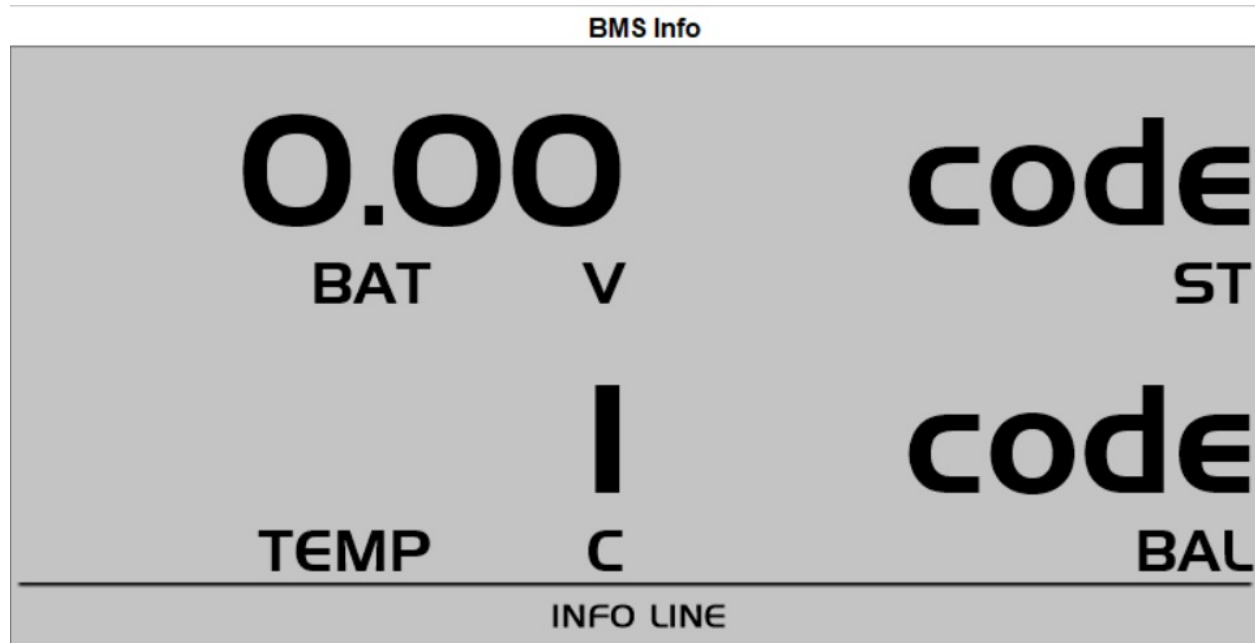


Figura 7. Hoja de visualización general del BMS.

Capítulo 7

Implementación software

En este capítulo se describe la implementación software del sistema de gestión eficiente de baterías desarrollado para la motocicleta eléctrica de competición. El sistema ha sido programado en lenguaje C++ [18], utilizando el entorno de desarrollo de Arduino, y se ejecuta sobre los microcontroladores ESP32 integrados en los dos hardware BMS del vehículo. Para estructurar el código se ha adoptado una arquitectura basada en programación orientada a objetos, lo que ha permitido encapsular funcionalidades, mejorar la legibilidad del código y facilitar su mantenimiento y ampliación.

Aunque no se incluye un análisis del código, se describen las estructuras lógicas y funcionales principales que componen la implementación. Para ello, se emplean diagramas de bloques y de flujo que permiten comprender el comportamiento general del sistema.

7.1. Funcionamiento general

El sistema de gestión eficiente de baterías debe cumplir con las funciones propuestas: la monitorización del paquete de baterías, la protección ante condiciones anómalas, el balanceo de celdas y el control automático del proceso de carga. Para implementar estas tareas, se ha desarrollado la lógica de funcionamiento representada en la Figura 1. Este esquema resume de forma simplificada y general el comportamiento global del sistema.

A partir de esta visión general, el capítulo se desarrolla abordando cada bloque funcional de manera independiente, con el objetivo de comprender su lógica interna sin necesidad de

analizar directamente el código programado.

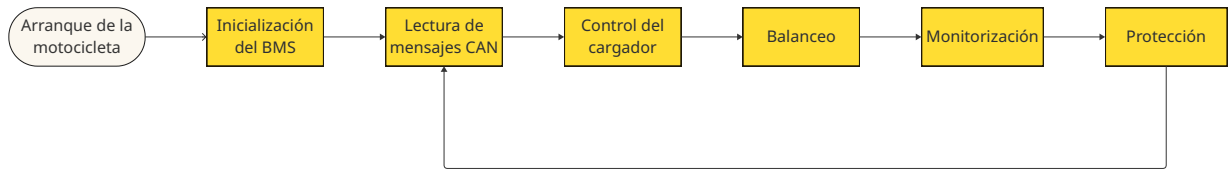


Figura 1. Diagrama general del funcionamiento.

El funcionamiento general del sistema (Figura 1) comienza con el arranque de la motocicleta. En ese momento se alimenta toda la electrónica del vehículo, incluyendo los microcontroladores ESP32 sobre los que se implementan las funciones del sistema de gestión de baterías. Al recibir alimentación, los ESP32 se inician automáticamente y comienzan la ejecución del programa.

El código arranca con una fase de inicialización del sistema BMS y, una vez completada, entra en un bucle principal que se ejecuta de forma indefinida hasta que se apaga la motocicleta. En cada iteración del bucle se realizan, de manera secuencial, la lectura de mensajes CAN, el control del cargador, el balanceo de celdas, la monitorización de la batería y su protección. Al finalizar estas tareas, el sistema vuelve al inicio del ciclo, comenzando nuevamente con la lectura de mensajes.

Una característica clave de esta implementación es que el código ha sido diseñado bajo un esquema no bloqueante. Esto implica que ninguna función detiene la ejecución general del programa durante un tiempo indefinido. Todas las tareas se ejecutan de forma continua, sin esperas prolongadas, lo que permite que el sistema responda en tiempos muy reducidos, normalmente inferiores a unas decenas de milisegundos por ciclo completo. Esta capacidad de respuesta es esencial, ya que los mensajes CAN llegan al sistema de forma asíncrona y deben ser atendidos con inmediatez. Si el tiempo entre iteraciones del bucle es demasiado elevado, el búfer del controlador MCP2515, encargado de recibir los mensajes CAN, puede saturarse y provocar pérdida de información. Este tipo de fallos comprometería la comunicación entre

módulos y, con ello, el correcto funcionamiento del sistema de gestión de baterías.

7.2. Inicialización del BMS

La inicialización del sistema BMS se ejecuta una sola vez y se lleva a cabo en una secuencia de cuatro etapas, tal como se representa en la Figura 2.



Figura 2. Diagrama de inicialización del BMS.

En primer lugar, se realiza la inicialización de variables, donde se crean los objetos necesarios y se definen las estructuras internas del programa. Esta fase incluye las configuraciones típicas del entorno de ejecución y la puesta a cero de contadores, indicadores y registros internos.

A continuación, se inicializa el circuito integrado bq76PL455A-Q1 mediante la ejecución de la función `InitializeBMS()`.

En la tercera fase se configura la interfaz de comunicación CAN, ajustando el MCP2515 a una velocidad de 500 kbit/s y un reloj de 8 MHz. Este paso es fundamental para permitir la comunicación entre los distintos módulos del sistema.

Finalmente, se introduce una espera con el objetivo de garantizar que todos los componentes hayan quedado correctamente configurados antes de entrar en el ciclo principal. El arranque se indica visualmente mediante el parpadeo de varios LEDs, lo que permite comprobar que la inicialización se está llevando a cabo correctamente.

7.3. Lectura de mensajes CAN

Este bloque constituye una parte fundamental del sistema, ya que se encarga de leer el mensaje almacenado en el búfer del MCP2515 y procesarlo según su CAN ID (Figura 3). En esta etapa también se implementa la detección automática de conexión y desconexión del cargador, fundamental para la gestión de la carga automática.

Los mensajes CAN recibidos se clasifican en función del módulo que lo ha enviado. Los provenientes de la botonera permiten controlar funcionalidades como la activación o desactivación del balanceo y controlar el proceso de carga. Los mensajes enviados por otro ESP32 del sistema BMS permiten conocer la información del paquete completo. El controlador proporciona la tensión que reconoce de la batería, la cual es la tensión que se supervisa en la carga automática.

7.3.1. Lectura de mensajes CAN del cargador

El cargador es el módulo que más tipos de mensajes maneja. Tal como se detalla en la Figura 4, existen tres tipos principales de mensajes CAN relacionados con él: mensajes de configuración, mensajes de precarga y un mensaje informativo que contiene la tensión y la corriente de salida del cargador en cada instante. Este último mensaje se transmite con una frecuencia de un segundo siempre que el cargador permanece conectado al sistema, y se emplea para detectar su presencia. Para ello se ejecuta la función `procesarCan()`.

La recepción de estos mensajes resulta esencial para el funcionamiento correcto del sistema. En caso de no recibir alguno de ellos, no es posible continuar con el proceso de carga. Si el cargador no responde durante la etapa de configuración o precarga, permanece bloqueado en ese estado hasta que se supera un tiempo de espera predefinido, momento en el que se reinicia la carga desde el estado de inicio. De forma similar, si el mensaje informativo deja de recibirse durante la carga, el sistema deduce que el cargador ha sido desconectado. En ambos casos, la lógica del BMS reinicia automáticamente el estado del cargador y se prepara para reanudar la comunicación si se detecta de nuevo la conexión en el bus CAN.

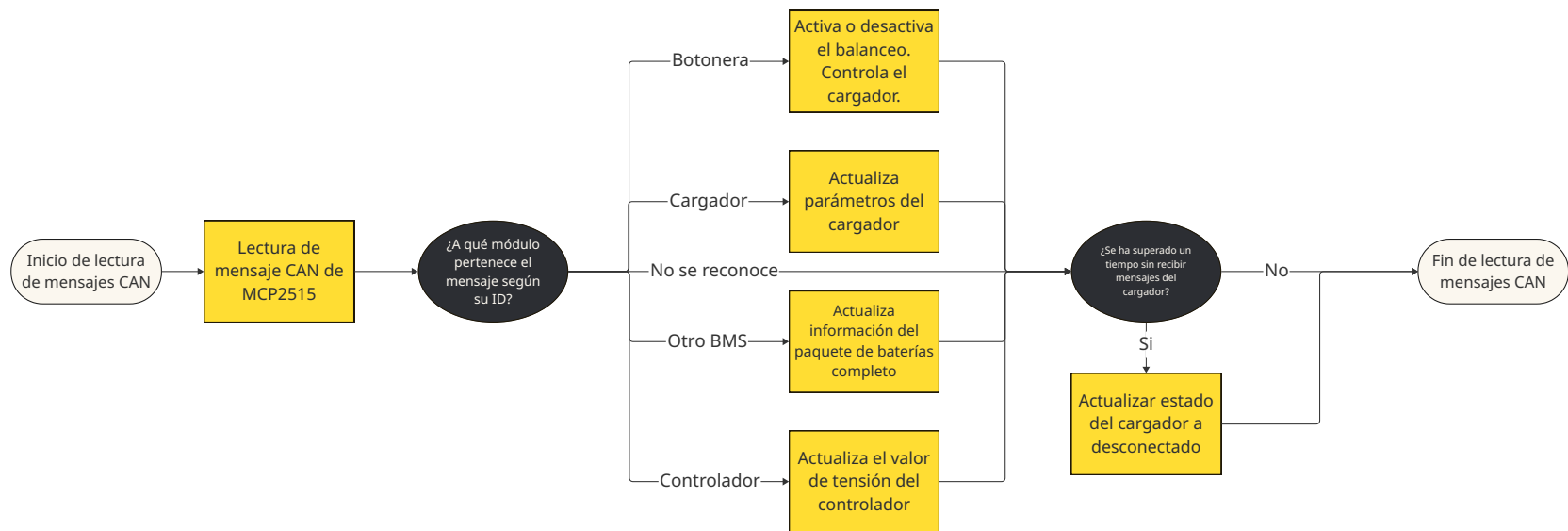


Figura 3. Diagrama general de lectura de mensajes CAN.

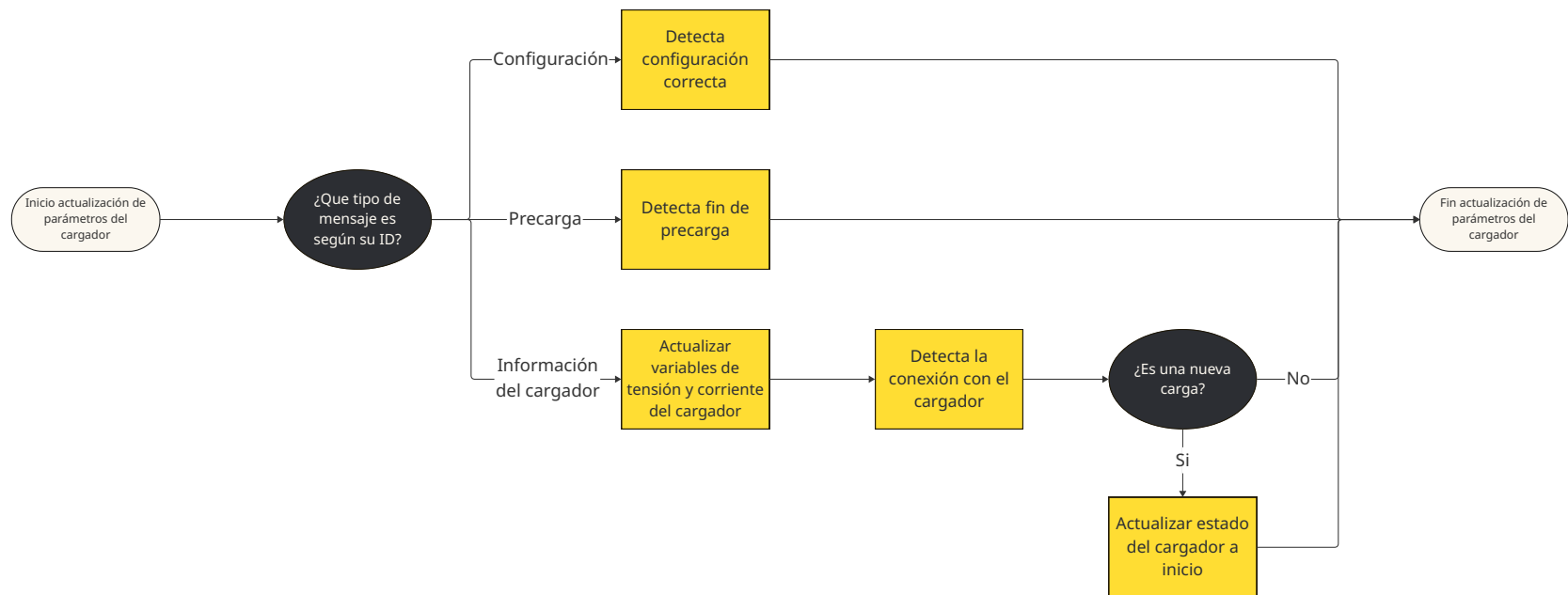


Figura 4. Diagrama de lectura de mensajes CAN del cargador.

7.4. Control del cargador

El comportamiento del cargador se ha implementado como una máquina de estados, tal como se representa en la Figura 5. El sistema cambia entre los distintos estados a medida que se van cumpliendo las condiciones necesarias en cada uno de ellos, comprobaciones que se realizan a partir de la información recibida en la Lectura de mensajes CAN del cargador.

Los posibles estados que puede adoptar el cargador son los siguientes: desconectado, inicio, configuración, precarga, carga_CC, carga_CV y carga_fin.

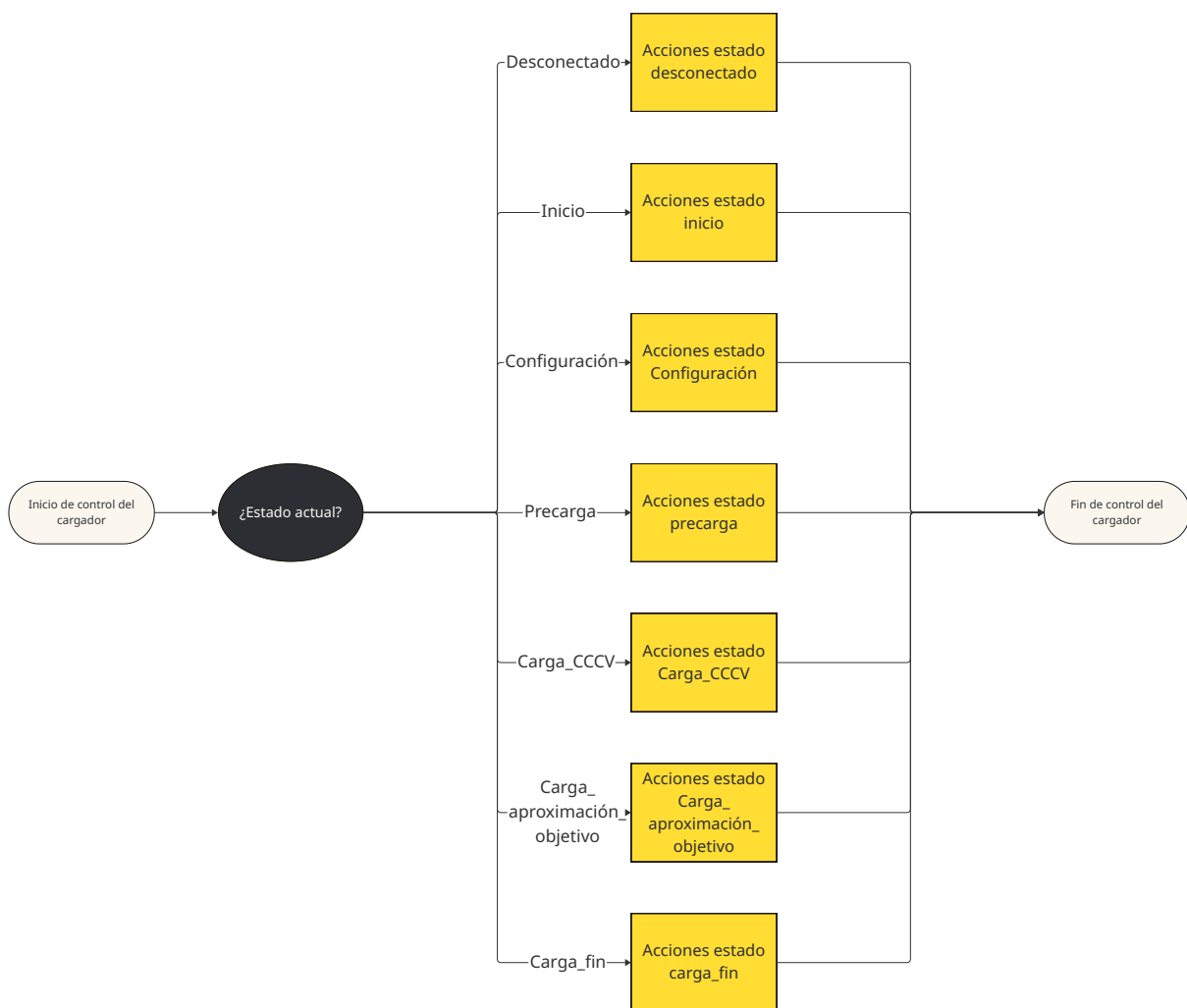


Figura 5. Diagrama de la máquina de estados del cargador.

7.4.1. Estado desconectado del cargador

En este estado (Figura 6) se ejecuta únicamente la función descrita en el apartado comprobar_conexion(), encargada de comprobar si el cargador se ha conectado a la motocicleta. Cuando se detecta la conexión, se actualiza el estado del cargador a inicio para dar paso al proceso de carga.

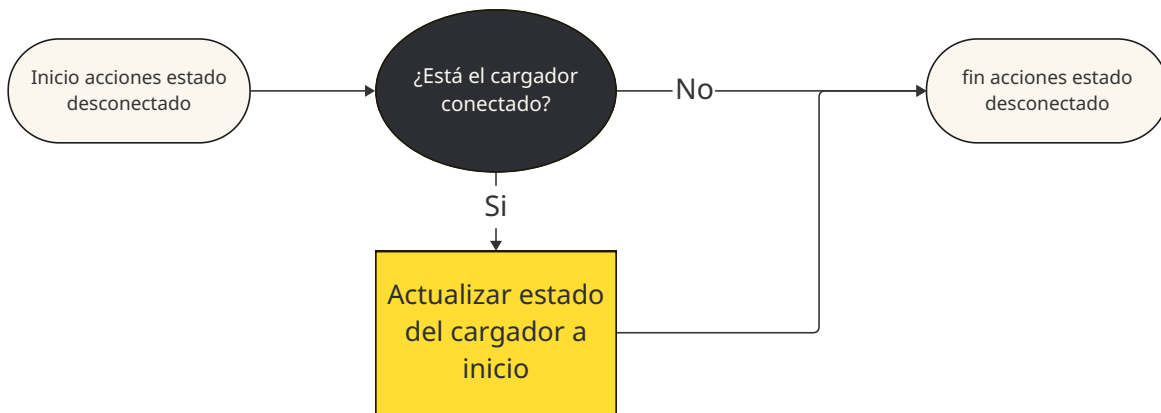


Figura 6. Diagrama del estado desconectado del cargador.

7.4.2. Estado inicio del cargador

En este estado (Figura 7) se envía la configuración al cargador mediante la función descrita en el apartado enviar_configuracion_inicial(). Una vez completado el envío, se actualiza el estado a configuración para continuar con el proceso de carga.

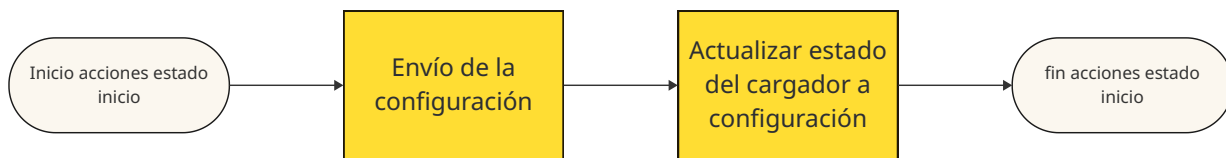


Figura 7. Diagrama del estado inicio del cargador.

7.4.3. Estado configuración del cargador

En este estado (Figura 8) se comprueba si la configuración se ha realizado correctamente mediante la función `comprobar_configuracion()`. Si la comprobación es correcta, se envía la orden de precarga con la función `enviar_precarga()` y se actualiza el estado del cargador a precarga. Además, este estado tiene un timeout que permite detectar si se ha perdido el mensaje de configuración. En ese caso, se reinicia el proceso y se vuelve al estado de inicio.

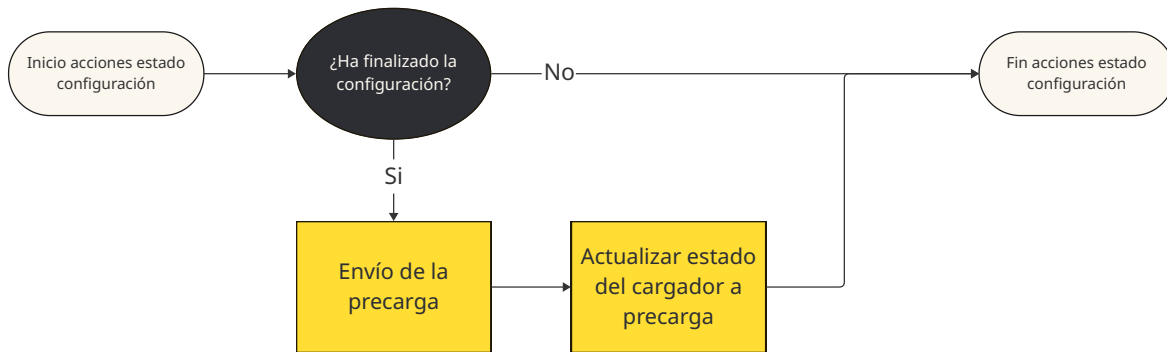


Figura 8. Diagrama del estado configuración del cargador.

7.4.4. Estado precarga del cargador

En este estado (Figura 9) se comprueba si la precarga se ha realizado correctamente mediante la función `comprobar_precarga()`. Si la comprobación es correcta, se envía la orden de finalizar la precarga con la función `finalizar_precarga()` y se actualiza el estado del cargador a `carga_CC`. Además, este estado tiene un timeout que permite detectar si se ha perdido el mensaje de precarga. En ese caso, se reinicia el proceso y se vuelve al estado de inicio.

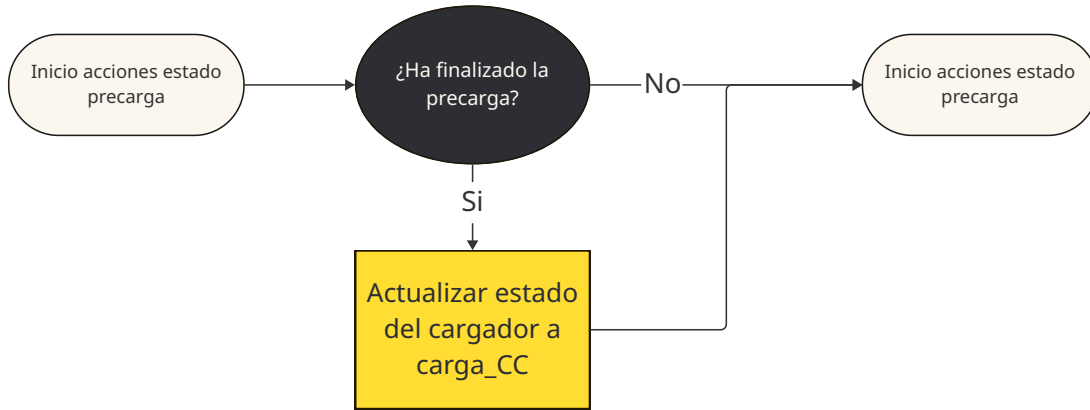


Figura 9. Diagrama del estado precarga del cargador.

7.4.5. Estado carga_CC del cargador

Este estado (Figura 10) es en el que el cargador permanece la mayor parte del tiempo durante una carga normal. En él se lleva a cabo la carga de la batería a corriente constante mediante la función `ejecutar_carga_CC()`.

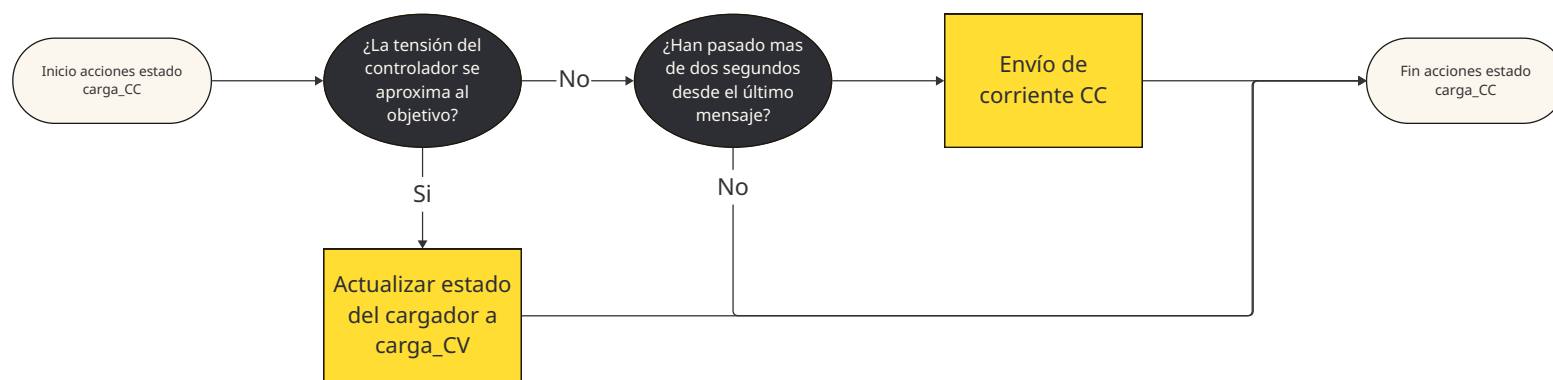


Figura 10. Diagrama del estado carga_CC del cargador.

7.4.6. Estado carga_CV del cargador

En este estado (Figura 11), el cargador realiza la etapa de carga a tensión constante mediante la función `ejecutar_carga_CV()`.

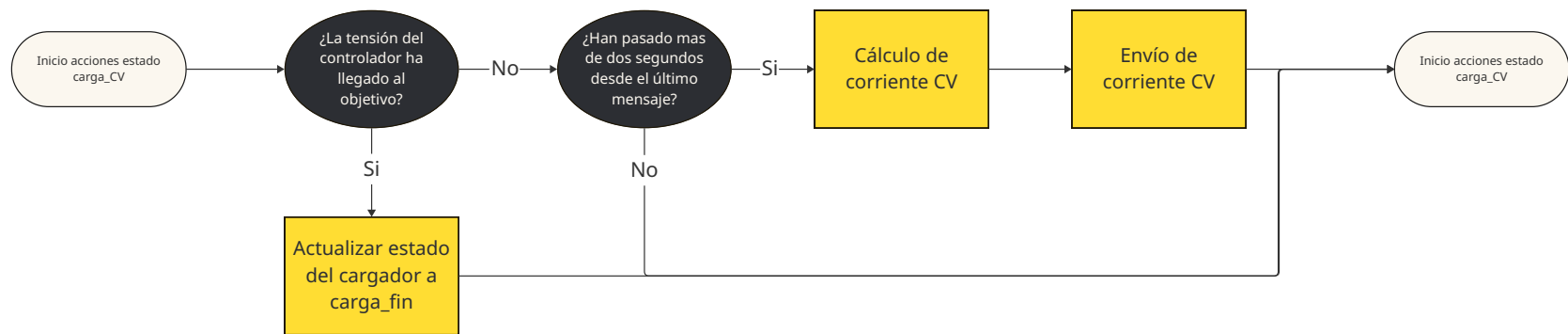


Figura 11. Diagrama de carga_CV del cargador.

7.4.7. Estado carga_fin del cargador

En este estado (Figura 12), el cargador se detiene ejecutando la función `detener_cargador()`.

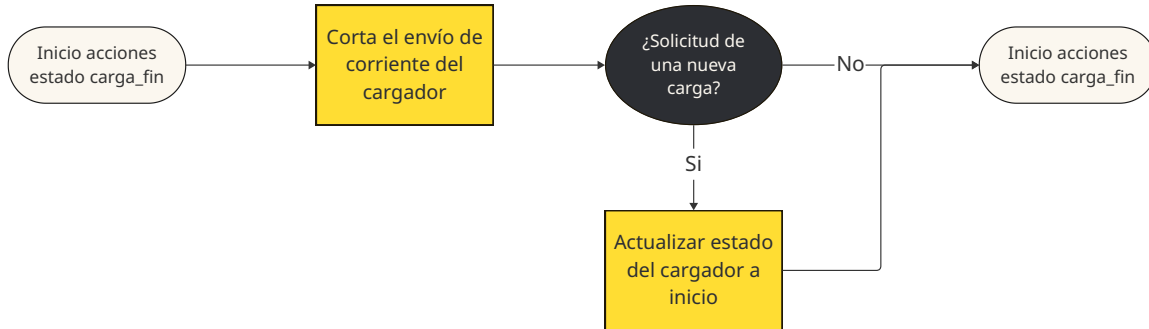


Figura 12. Diagrama del estado carga_fin del cargador.

7.5. Balanceo

En esta motocicleta de competición, el balanceo no se activa de forma automática, sino manualmente mediante la botonera (Figura 13). Esta decisión permite que los ingenieros responsables del mantenimiento y preparación de la batería tengan un control total sobre el estado del paquete, lo que facilita alcanzar los máximos niveles de rendimiento del sistema.

Aunque en esta aplicación no se implemente un funcionamiento automático, el sistema BMS cuenta con la lógica necesaria para detectar cuándo el balanceo ha finalizado correctamente. Por tanto, habilitar una activación automática sería una cuestión de definir en qué momento debe iniciarse el balanceo, en función de los requisitos particulares del vehículo en el que se utilice este sistema.

El balanceo está condicionado por la monitorización del paquete de baterías (Figura 13). Cada vez que se toma una muestra, el balanceo debe haberse detenido durante al menos medio segundo. Este tiempo de espera evita que la caída de tensión provocada por el balanceo distorsione las medidas de las celdas y se obtengan valores incorrectos.

El algoritmo de balanceo se describe en el apartado Algoritmo de balanceo, y su funcionamiento se resume en la Figura 14. Cuando se cumplen las condiciones necesarias, se ejecuta la función `autobalance()`; en caso contrario, se desactiva el balanceo de todas las celdas.

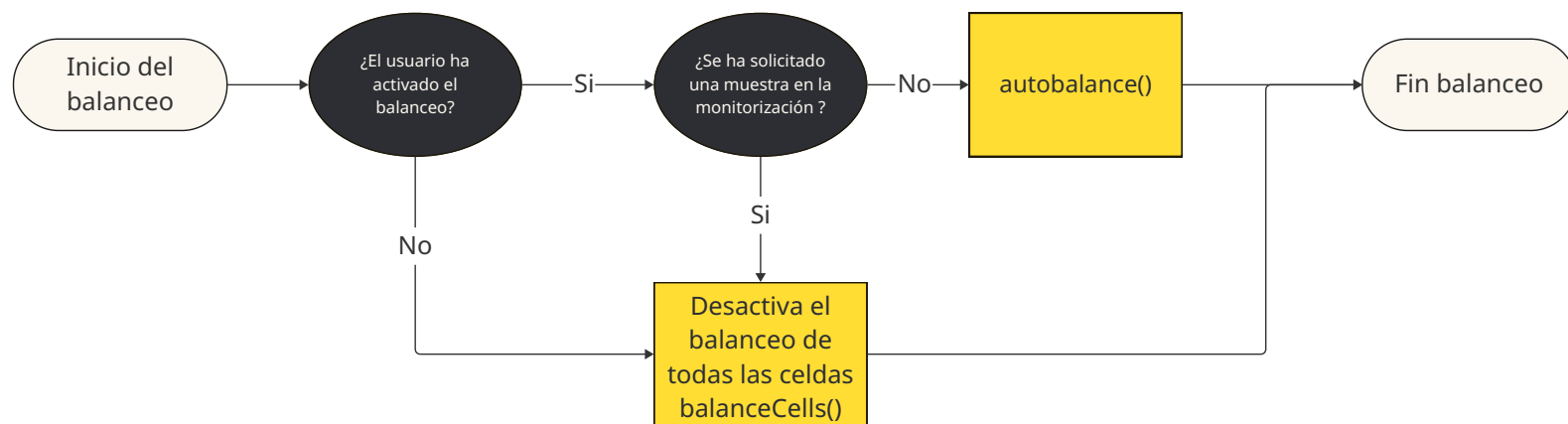


Figura 13. Diagrama del balanceo del BMS.

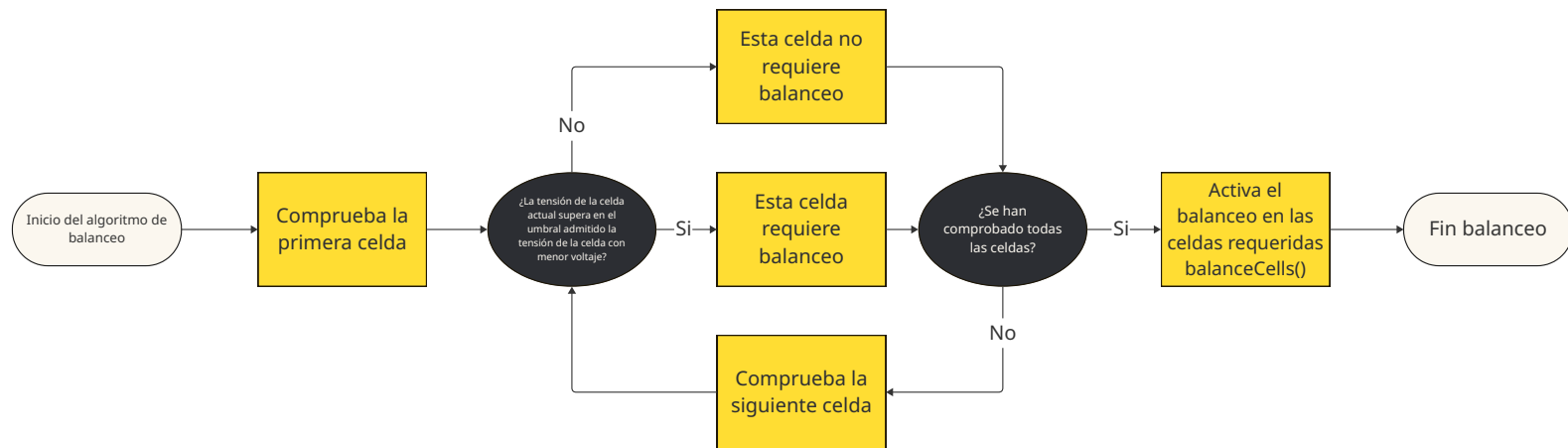


Figura 14. Diagrama del algoritmo de balanceo.

7.6. Monitorización

La monitorización es un proceso periódico cuyo tiempo de muestreo varía en función del estado del sistema. Este tiempo es especialmente relevante durante el balanceo, ya que cada vez que se toma una muestra, el balanceo debe permanecer desactivado durante al menos medio segundo. Por tanto, si el tiempo de muestreo se establece en un segundo, la mitad del tiempo se emplea en el balanceo y la otra mitad en realizar la medida. Para maximizar la eficacia del proceso, se utiliza un tiempo de muestreo de cuatro segundos cuando el balanceo está activo, y de medio segundo cuando no lo está.

En la Figura 15 se representa el diagrama de funcionamiento de la monitorización. Esta depende directamente del tiempo de muestreo: cuando se alcanza este intervalo, se solicita la toma de una muestra. Esta solicitud desactiva el balanceo, y una vez transcurrido al menos medio segundo con el balanceo apagado, se realiza la medida. A continuación, se gestionan los datos obtenidos y se anula la solicitud de muestra, permitiendo así reactivar el balanceo si es necesario.

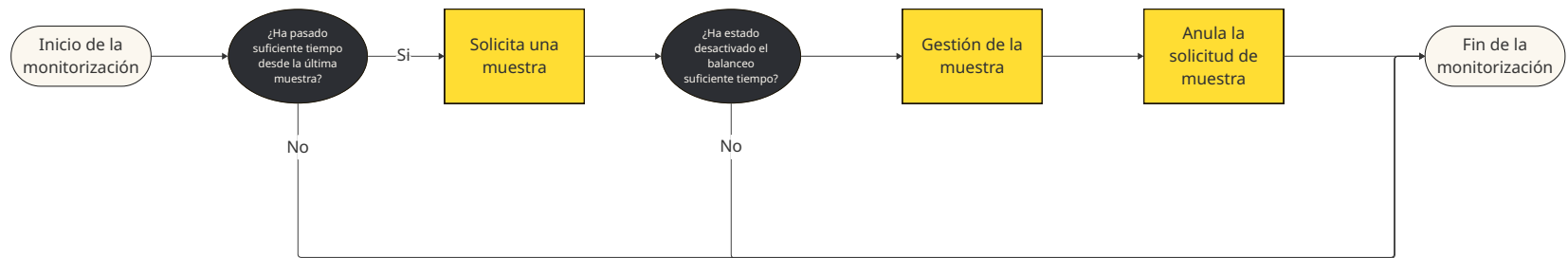


Figura 15. Diagrama de la monitorización del BMS.

El proceso de gestión de la muestra abarca más que la simple toma de datos. En primer lugar, se solicita al bq76PL455A-Q1 que capture una muestra y la almacene en su búfer interno mediante la función `RequestSample()`. A continuación, se transmite esa muestra al ESP32 a través de UART utilizando la función `ReadData()`. Los datos recibidos se procesan mediante la función `RawDataToProcessData()`, que los transforma en estructuras más manejables. Finalmente, los datos resultantes se envían a la pantalla mediante la función `CommunicationCANDisplay()` y al resto de nodos ESP32 del BMS mediante `sendInfoData()`.

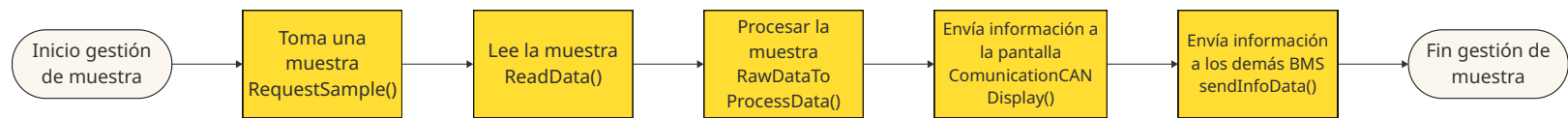


Figura 16. Diagrama de la gestión de muestra de la monitorización.

7.7. Protección

El sistema de protección se basa en la supervisión periódica de los valores de tensión y temperatura obtenidos durante la monitorización. Tras cada muestra, se comprueba que todas las variables se encuentren por debajo de unos umbrales máximos definidos. Si se cumplen estas condiciones, el sistema continúa funcionando con normalidad. En caso contrario, si alguna medida supera los límites permitidos, el BMS envía una orden de apertura a los contactores de la batería para cortar la tensión de forma segura. Este modo de funcionamiento seguro es eficaz para proteger tanto al sistema como al entorno. No obstante, durante la carrera no se activa esta protección automática, ya que una desconexión repentina de la batería a alta velocidad supondría un riesgo mayor para el piloto que el propio fallo térmico del paquete de baterías. Por este motivo, durante la carrera el sistema de protección solo advierte al piloto a través de la pantalla.

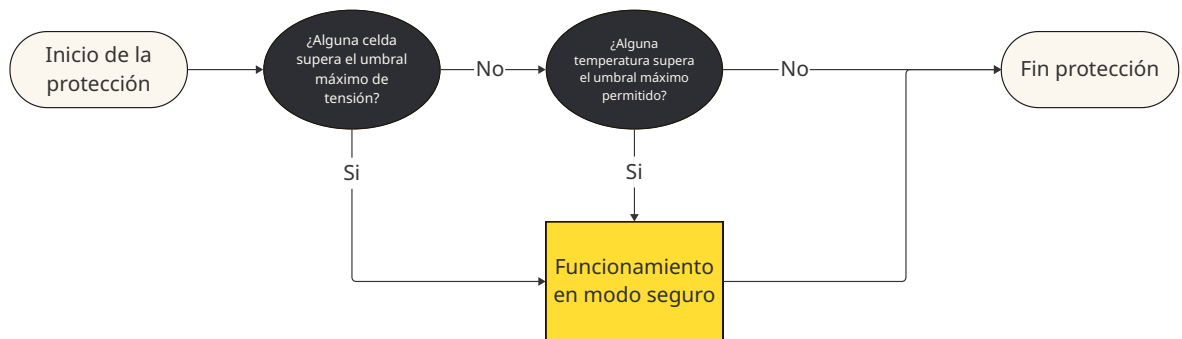


Figura 17. Diagrama de la protección del BMS.

Capítulo 8

Resultados

Este capítulo presenta los resultados obtenidos tras la validación del sistema en una motocicleta real de competición. Se analiza el comportamiento del sistema en las distintas funcionalidades incorporadas: monitorización, balanceo, carga, protección e interfaz de usuario.

8.1. Resultados de la monitorización

La funcionalidad de monitorización ha sido verificada mediante la visualización de datos en la pantalla Aim MXm. En la Figura 1 y Figura 2 se observa la lectura correcta de las tensiones individuales de las treinta celdas del sistema, divididas en dos bloques de quince celdas.

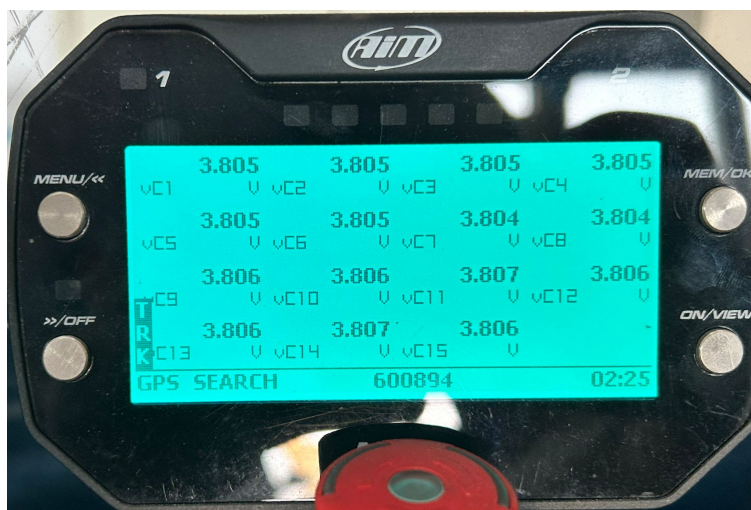


Figura 1. Visualización de las primeras quince celdas.

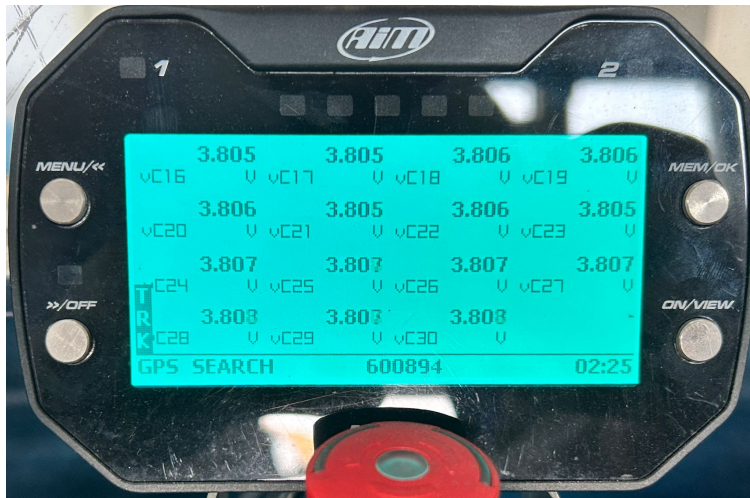


Figura 2. Visualización de las últimas quince celdas.

Las lecturas de temperatura se visualizan correctamente, tanto al inicio como tras una hora de funcionamiento. En las Figuras 3 y 4 se observa un aumento en la temperatura registrada por los sensores debido al calentamiento natural del sistema durante el uso.



Figura 3. Visualización de la temperatura al inicio del uso.



Figura 4. Visualización de la temperatura tras una hora de funcionamiento.

8.2. Resultados del balanceo

El sistema de balanceo pasivo ha sido evaluado a partir de un estado inicial de desbalanceo forzado, con un error máximo de 15 mV entre celdas, como se muestra en las figuras 5 y 6.

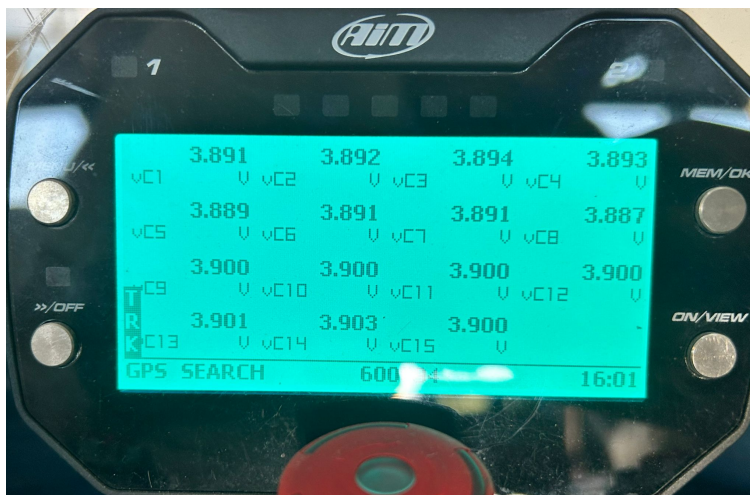


Figura 5. Estado inicial de balanceo de las primeras quince celdas.

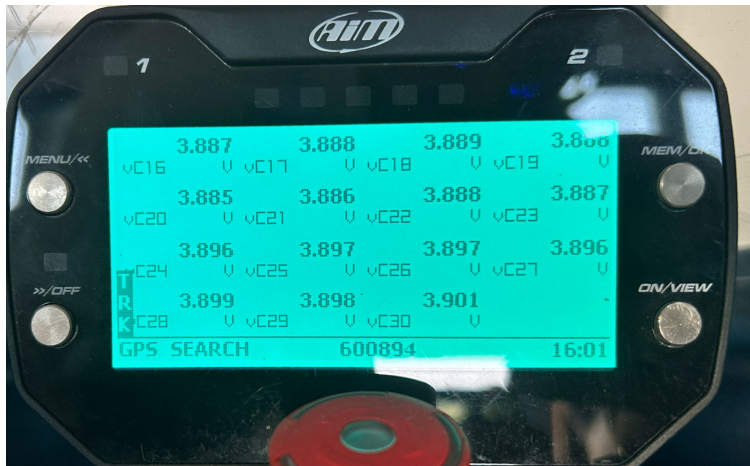


Figura 6. Estado inicial de balanceo de las últimas quince celdas.

Tras aproximadamente 16 horas, el error se reduce a 9 mV, como reflejan las Figuras 7 y 8. El funcionamiento de las celdas que se balancean se observa a través de los LEDs encendidos de las placas (Figura 9).

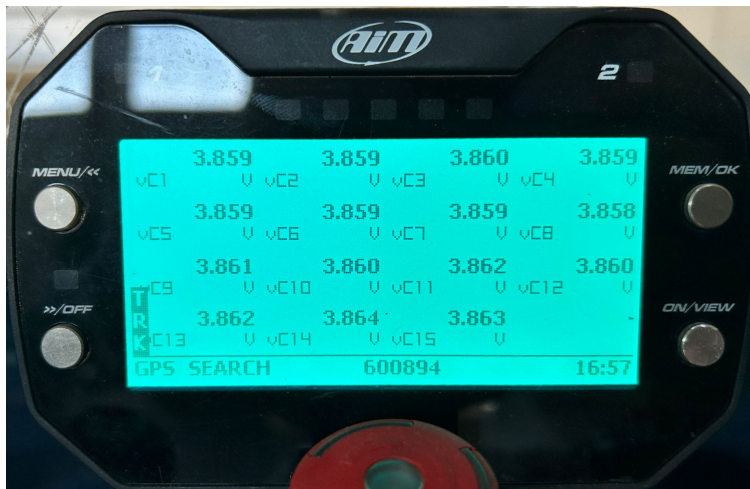


Figura 7. Balanceo intermedio de las primeras quince celdas.

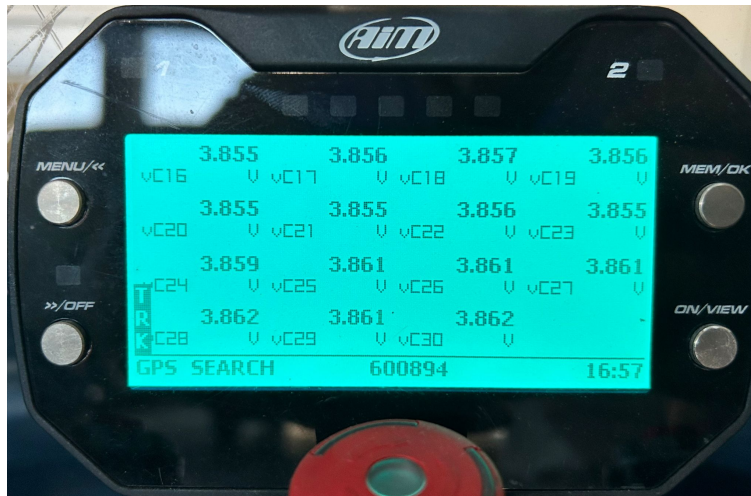


Figura 8. Balanceo intermedio de las últimas quince celdas.

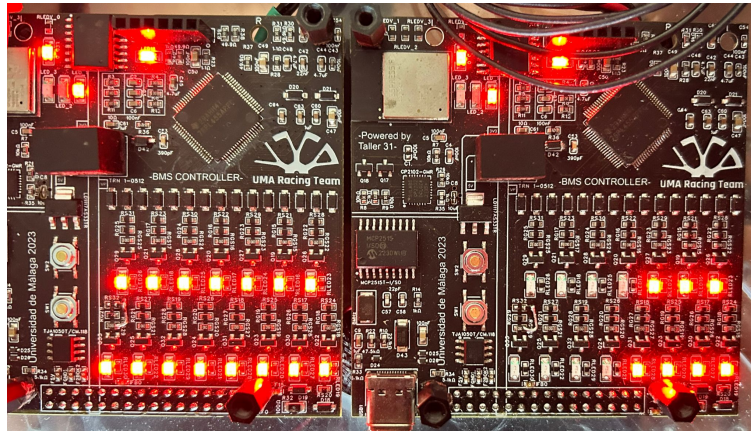


Figura 9. Visualización del balanceo en un punto intermedio.

Después de aproximadamente 40 horas (Figuras 10 y 11), el error máximo entre celdas se reduce a 2 mV, completando así el proceso de balanceo. El resultado es lento y ha requerido reducir en una décima de voltio la tensión de cada celda durante la ejecución del algoritmo. Este comportamiento se debe a que la placa actual emplea las resistencias de 75Ω en el circuito de balanceo y no las resistencias mejoradas de 16Ω . Aunque la mejora ya está proyectada, esta no se implementará hasta la próxima versión del hardware. Se espera que este cambio reduzca el tiempo de equilibrado a una quinta parte y disminuya significativamente la energía disipada durante el balanceo.

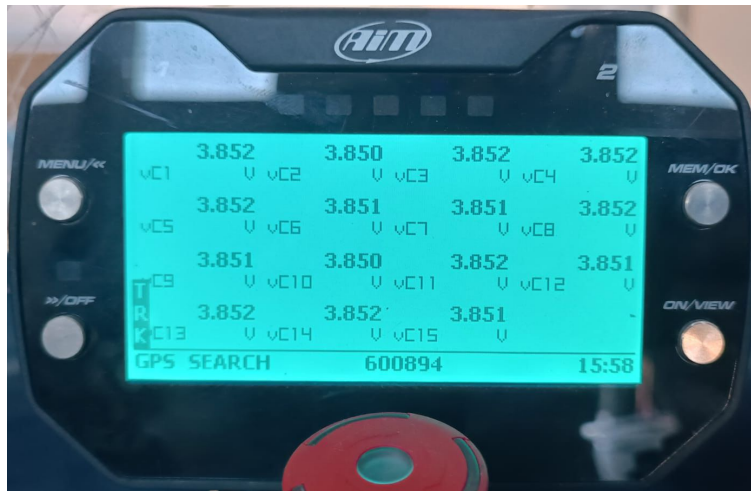


Figura 10. Visualización del balanceo final.

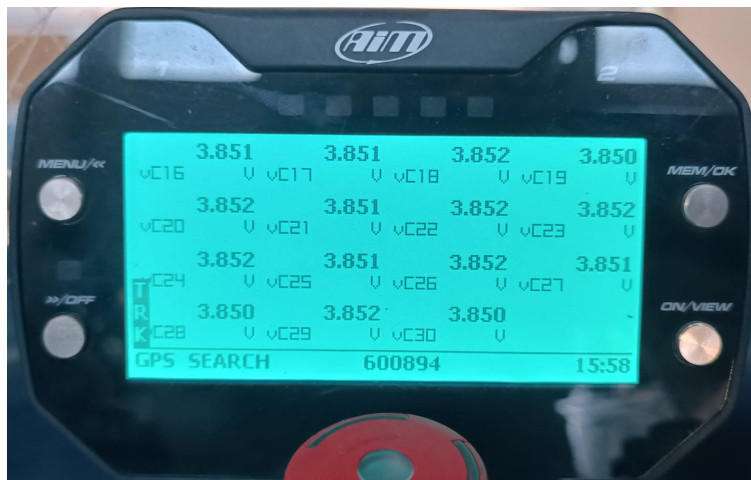


Figura 11. Visualización del balanceo final.

8.3. Resultados de la carga

La funcionalidad de control de carga ha sido validada a través de una carga programada desde 115,26 V hasta 117,2 V. El sistema detecta automáticamente el inicio de la carga y ejecuta correctamente el perfil CC-CV. La Figura 1 presenta el perfil teórico de este proceso, mientras que las figuras 12 y 13 muestran los resultados reales.

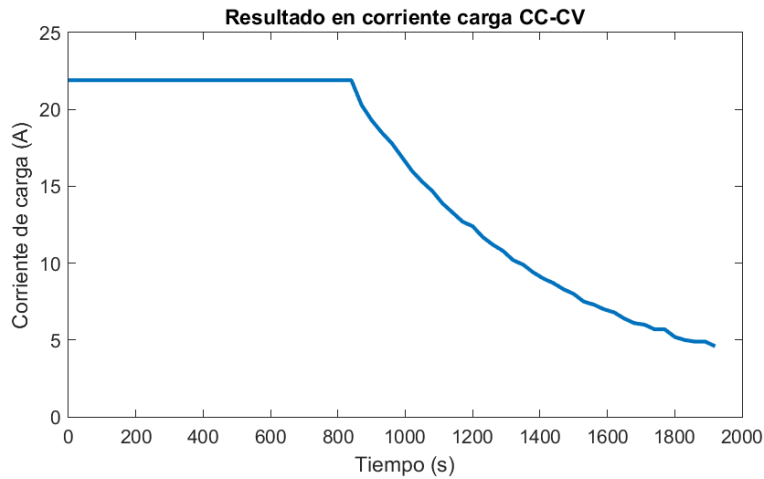


Figura 12. Evolución de la corriente suministrada durante la carga.

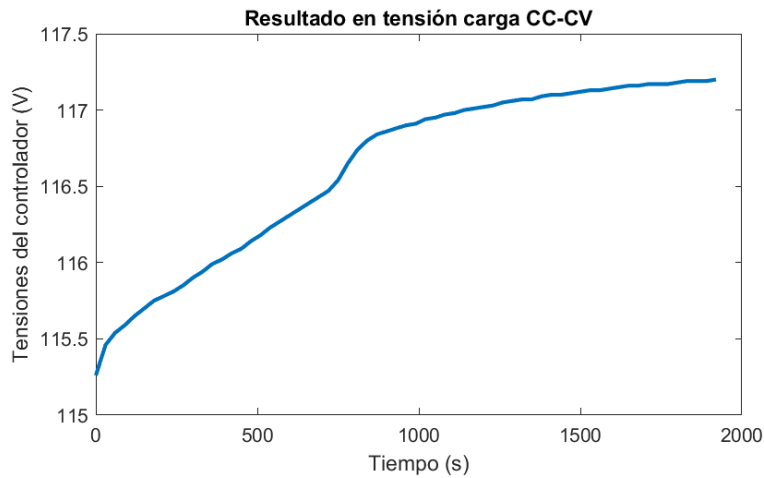


Figura 13. Evolución de la tensión medida durante la carga.

Cabe destacar que la curva de tensión mostrada en la Figura 13 no presenta una apariencia completamente plana durante la fase de tensión constante. Esto se debe a que la carga total programada en este caso es de aproximadamente 2 V, mientras que la zona correspondiente al tramo CV implica una variación de 0,5 V, lo que se traduce en un cambio inferior a 2 mV por celda. En una carga típica, como la que se realiza desde 105 V hasta 127,5 V, esta variación es aún menos perceptible. Al representar esta misma evolución con un eje Y que cubra el rango completo de carga, como se muestra en la Figura 14, se aprecia una mayor estabilidad en la fase de tensión constante, alineándose con el comportamiento esperado del perfil CC-CV.

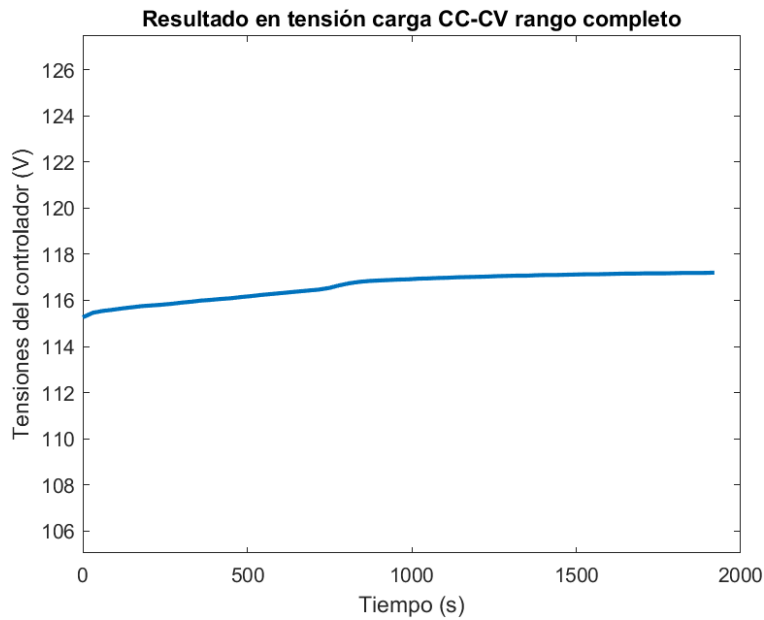


Figura 14. Evolución de la tensión del controlador durante una carga real con perfil CC-CV con eje reescalado.

La tensión final detectada por el sistema se muestra en el mensaje CAN representado en la Figura 15. Los bytes 0x07 y 0x50 indican un valor que, tras aplicar el factor de conversión, se traduce en 117,0 V. El error con respecto al objetivo es de tan solo 0,2 V, considerado admisible.

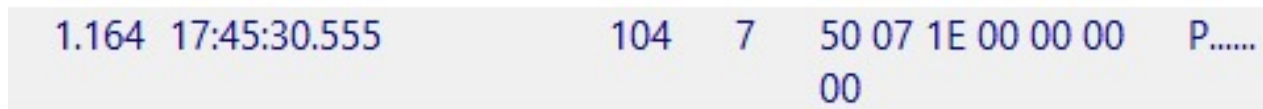


Figura 15. Evolución de la tensión del controlador durante una carga real con perfil CC-CV con eje reescalado.

8.4. Resultados de la protección

La protección ante condiciones de sobretensión y sobret temperatura se ha validado introduciendo manualmente valores ficticios que superan los límites máximos permitidos. En ambos casos, el sistema ha reaccionado de forma inmediata, desactivando el contactor de la batería y garantizando así la seguridad del conjunto.

8.5. Resultados de la interfaz de usuario

Se ha verificado el correcto funcionamiento de la interfaz de usuario implementada mediante la botonera de la motocicleta. El sistema permite al usuario activar y desactivar manualmente el proceso de balanceo de celdas, así como iniciar y detener la carga de la batería sin necesidad de reprogramación. Además, se ha comprobado que es posible modificar la corriente de carga desde la propia botonera, permitiendo ajustar este parámetro según las necesidades operativas del momento.

Capítulo 9

Conclusiones y líneas futuras

Este capítulo finaliza el trabajo y sintetiza las ideas más importantes obtenidas tras completarlo. En primer lugar se describe la metodología seguida para fundamentar las decisiones de diseño. A continuación se presentan los logros alcanzados, se analizan las limitaciones detectadas durante su implementación y, por último, se indican las líneas de trabajo futuro que permitirán seguir mejorando el BMS desarrollado para la motocicleta eléctrica de competición.

9.1. Metodología

Para alcanzar los objetivos del proyecto se han estudiado en profundidad las características de las baterías de ion-litio, el funcionamiento general de los sistemas BMS y las curvas de carga más habituales. También se ha analizado detalladamente el funcionamiento e integración de los elementos específicos utilizados en la motocicleta, como el circuito integrado bq76PL455A-Q1, el cargador Zivan NG3 y la pantalla AIM MXm. Este análisis ha permitido tomar decisiones de diseño fundamentadas y alineadas con los requisitos de la competición, maximizando el aprovechamiento de los recursos disponibles y hacer funcionar todos los recursos.

9.2. Logros alcanzados

A continuación se describen los logros técnicos obtenidos como resultado de cumplir los objetivos específicos planteados en el proyecto.

En primer lugar, se ha reestructurado por completo el código existente, migrando a una arquitectura orientada a objetos. Esta reorganización ha permitido encapsular las funcionalidades de monitorización ya presentes, sentando una base sólida para la incorporación de futuras implementaciones.

A partir de esta base, se ha diseñado e implementado un algoritmo de balanceo pasivo robusto y funcional. Durante su desarrollo se han detectado errores en el hardware original, que han sido corregidos, y se han propuesto mejoras estructurales para la próxima versión del circuito. El algoritmo permite alcanzar una precisión final de 2 mV entre celdas, un resultado que compite con sistemas de gestión de baterías comerciales.

También se ha desarrollado la comunicación automática con el cargador Zivan NG3 mediante el protocolo CAN. Esta implementación permite que el BMS detecte de forma autónoma la conexión del cargador, configure los parámetros de seguridad, lleve a cabo la precarga y ejecute una carga controlada con perfil CC-CV, adaptada a la tensión objetivo del controlador de la motocicleta para maximizar su eficiencia. Los resultados experimentales confirman que la carga se realiza correctamente, alcanzando la tensión programada con un error final de tan solo 0,2 V, valor que se considera perfectamente aceptable.

Se ha integrado una capa de protección frente a condiciones de sobretensión y sobret temperatura. El sistema supervisa condiciones de sobretensión y sobret temperatura y actúa sobre el contactor de la batería para interrumpir el flujo de corriente de forma segura, preservando la integridad del sistema eléctrico del vehículo.

Además, se ha incorporado una interfaz de usuario mediante la botonera de la motocicleta. Esta permite iniciar y detener manualmente tanto el proceso de carga como el balanceo, así como ajustar la corriente de carga, todo ello sin necesidad de reprogramar el sistema.

Por último se ha logrado que todas estas funcionalidades operen de forma simultánea en un sistema embebido. Para ello, se ha desarrollado un protocolo de comunicaciones propio sobre la red CAN de la motocicleta y se ha implementado una lógica no bloqueante en el código,

que permite gestionar eficazmente la concurrencia de tareas y mantener la sincronización entre módulos.

El conjunto de logros alcanzados representa un avance sustancial respecto al sistema anterior, que se limitaba a funciones básicas de monitorización. A partir de esa base inicial, se ha construido un sistema completo, capaz de gestionar activamente la carga, equilibrar las celdas, proteger la batería frente a condiciones anómalas y comunicarse con el resto de módulos del vehículo en tiempo real. El nuevo sistema no solo mejora el rendimiento y la seguridad del conjunto, sino que establece una base sólida y escalable para futuras versiones del BMS en el contexto de la competición. Su implementación en la motocicleta representa un salto de calidad en la gestión de baterías del vehículo, alineado con las exigencias técnicas de MotoStudent.

9.3. Limitaciones

A pesar de los logros alcanzados, el sistema presenta algunas limitaciones relevantes que conviene destacar.

En primer lugar, el proceso de balanceo pasivo resulta excesivamente lento y supone una disipación energética elevada. Esta limitación está directamente relacionada con el uso de resistencias de $75\ \Omega$ en el circuito de balanceo, que restringen la corriente máxima a aproximadamente $56\ \text{mA}$ por celda. Para solventarlo, ya se ha proyectado una mejora basada en resistencias de $16\ \Omega$, que permitirá multiplicar por cinco la corriente de balanceo y reducir de forma significativa tanto el tiempo necesario como la energía disipada.

Por otro lado, el sistema no realiza una estimación directa del estado de carga de la batería, sino que basa su gestión en las tensiones individuales de celda.

Por último, aunque el sistema dispone de mecanismos de protección para la batería frente a condiciones de riesgo, estas funciones se deshabilitan durante la carrera para priorizar la seguridad del piloto. Esto implica que, en condiciones de carrera, la protección depende de la supervisión manual.

9.4. Líneas futuras

Como continuación natural de este trabajo, se plantea validar el sistema desarrollado en un entorno real de alta exigencia, concretamente durante la participación en la próxima edición de la competición MotoStudent 2025. La implementación del BMS en la motocicleta de carrera permitirá comprobar su fiabilidad y comportamiento bajo condiciones reales, donde su correcto funcionamiento resulta determinante en los resultados de la competición.

Una de las principales tareas pendientes consiste en verificar experimentalmente la mejora proyectada en el circuito de balanceo mediante el uso de resistencias de $16\ \Omega$. Se espera que esta modificación reduzca sustancialmente el tiempo de balanceo y mejore la eficiencia energética del proceso.

Asimismo, se pretende optimizar la precisión del sistema de carga. En particular, se buscará reducir el error final en la carga CC-CV desde los actuales $0,2\ \text{V}$ hasta el orden de las centésimas, elevando así la eficiencia y la repetibilidad en condiciones de competición.

Otra mejora prevista es la implementación de un sistema de carga en paralelo utilizando dos cargadores Zivan NG3, con el objetivo de disminuir el tiempo total de carga.

Por último, se contempla la incorporación de un método de estimación del estado de carga basado en técnicas de coulomb counting, complementando el enfoque actual basado únicamente en la tensión de celda. Esto permitiría una gestión más precisa del sistema energético, especialmente en regímenes de carga y descarga variables.

Bibliografía

- [1] Toll, M., *DIY Lithium Batteries: How to Build Your Own Battery Packs*. CreateSpace Independent Publishing Platform, 2017.
- [2] Yang, S., Liu, X., Li, S., y Zhang, C., *Advanced Battery Management System for Electric Vehicles. Key Technologies on New Energy Vehicles*, Singapore: Springer Nature Singapore Pte Ltd., 2023, doi:10.1007/978-981-19-3490-2.
- [3] Liu, Y.-H., Teng, J.-H., y Lin, Y.-C., “Search for an optimal rapid charging pattern for lithium-ion batteries using ant colony system algorithm,” *IEEE Transactions on Industrial Electronics*, vol. 52, no. 5, pp. 1328–1336, 2005, doi:10.1109/TIE.2005.855672.
- [4] Andrea, D., *Battery Management Systems for Large Lithium Ion Battery Packs*. Norwood, MA, USA: Artech House, 2010.
- [5] Andrea, D., “Balance — battery management systems,” 2019, <https://lilionbms.com/balance/>. Último acceso: 16 de junio de 2025.
- [6] Corrigan, S., *Introduction to the Controller Area Network (CAN)*. Texas Instruments, 2016. Application Report SLOA101B.
- [7] Electric Drive Engineering, “Sevcon gen4 size 6 80v,” 2025, <https://electricdriveengineering.com.au/product/sevcon-gen4-size-6-80v/>. Accedido el 16 de junio de 2025.
- [8] Texas Instruments, *bq76PL455EVM Evaluation Module User’s Guide*. Texas Instruments, 2015. SLUUB45A – February 2014 – Revised May 2015.
- [9] Texas Instruments, *bq76PL455A-Q1: Precision Battery Monitor and Secondary Protector*. Texas Instruments, 2022. SLUSC54C – January 2015 – Revised March 2022.

- [10] Holland, S., bq76PL455A-Q1 Software Design Reference. Texas Instruments, 2015. Application Report SLVA617A, November 2014 – Revised May 2015.
- [11] De la Vega, R., Andréu Lamperti, S., Rossi, S., Spina, M., Brizzio, N., y Leegstra, R., “Propuesta y comparación de algoritmos de balanceo pasivo de celdas nmc para una batería de iones de litio,” *Revista Politécnica*, vol. 55, pp. 83–94, 2025, doi:10.33333/rp.vol55n1.08.
- [12] Ewert Energy Systems, “Description of balance parameter.” https://www.orionbms.com/manuals/utility/param_balancing_description.html, 2025. Accedido el 16 de junio de 2025.
- [13] Zivan Srl, NG3 CANBUS Protocol Specification. Italy, 2021. Hoja de datos técnica del cargador Zivan NG3.
- [14] Zivan Srl, CB 1.08 CANBus Communication Protocol (Document D02003-03). Italy, 2017. Hoja de datos técnica del protocolo CAN para cargadores CB.
- [15] Pini, N., RE Firmware Manual English: CAN Instruction Messages – RE Curve – PWM CAN Bus Protocol. Zivan, 2015. Document D02023-06, revision 06, July 2015.
- [16] AiM Technologies, “Aim mxm — dash logger,” 2025, <https://www.aimtechnologies.com/mxm-tech-spec/>. Accedido el 16 de junio de 2025.
- [17] AiM Tech Srl, MXm User Guide. Release 1.06, 2021. Manual técnico del sistema de adquisición de datos MXm.
- [18] cplusplus.com, “Referencia del lenguaje c++,” 2025, <https://cplusplus.com/>. Accedido el 16 de junio de 2025.