

DISEÑO Y PROGRAMACIÓN DE UN SISTEMA DE GESTIÓN DE BATERÍAS (BMS) PARA BATERÍAS LiFePO_4

MÁSTER EN INGENIERÍA MECATRÓNICA

Autor: Marcos Ahulló Carbonell

Tutor UPV: Vicente Fermín Casanova Calvo

Tutor Robotnik SLL: Miguel Moreno Muñoz



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

MEMORIA DESCRIPTIVA	11
1. INTRODUCCIÓN:	12
1.1. REDACCIÓN DEL PROYECTO:	12
1.2. JUSTIFICACIÓN DEL PROYECTO:	13
1.3. OBJETIVOS DEL PROYECTO:	15
1.4. METODOLOGÍA A SEGUIR:	17
2. ESTUDIO PREVIO:	20
2.1. CONCEPTOS BÁSICOS SOBRE BATERÍAS:	20
2.2. TIPOS DE BATERÍAS:	21
- Baterías de plomo-ácido:	
- Baterías de Níquel-Cadmio (Ni-Cd):	
- Baterías de Níquel-Hidruro metálico (Ni-MH):	
- Baterías Ión-Litio (Li-Ion):	
- Baterías de Polímero de Litio Ferroso (LiFePO ₄):	
2.3. BATERÍAS EN FASE DE DESARROLLO:	25
- Baterías con nanotecnologías:	
- Baterías de litio en estado sólido:	
- Supercondensadores:	
- Baterías de metal-aire:	
2.4. PARÁMETROS CLAVE:	26
- Densidad energética:	
- Potencia específica:	
- Eficiencia de carga/descarga:	
- Ciclos de vida:	
- Velocidad de recarga:	
2.5. PROBLEMÁTICA CON LAS BATERÍAS LiFePO₄:	28
2.6. FUNCIONES DEL BMS COMO SOLUCIÓN:	31
- Maximizar las prestaciones de la batería:	
- Monitorizar la batería:	

- Estimar el estado de la batería:
- Proteger la batería:
- Exportar todos los datos:

2.7. TIPOS DE BMS EXISTENTES: 32

- Centralizado:
- Modular:
- Maestro-Esclavo:
- Distribuido:

2.8. ELECCIÓN DEL TIPO DE BMS PARA EL PROYECTO: 34

3. DISEÑO HARDWARE: 36

3.1. DIAGRAMA DE ESTADOS DEL BMS: 36

3.2. PARTES DEL BMS: 38

3.3. REQUISITOS DEL BMS: 39

3.4. TARJETA DE CONTROL: 41

- 3.4.1. Circuito de balanceo: 42
- 3.4.2. Cálculo y elección del MOSFET de balanceo: 43
- 3.4.3. Circuito de medidas: 46
- 3.4.4. Multiplexor y amplificador diferencial: 46
- 3.4.5. Configuración de las celdas: 48
- 3.4.6. Circuito de acondicionamiento de la señal: 49
- 3.4.7. Elección de los componentes de acondicionamiento: 50
- 3.4.8. El microcontrolador: 53
- 3.4.9. Elección del microcontrolador PIC: 55
- 3.4.10. Sensores de temperatura: 58
- 3.4.11. Elección de los sensores de temperatura: 59
- 3.4.12. Medios de comunicación: 59
 - Protocolo USB:
 - Interfaz I2C:
- 3.4.13. Resto de conexiones: 64

3.5. TARJETA DE POTENCIA: 66

3.5.1. Cálculo del MOSFET de potencia:	67
3.5.2. Sensor de corriente efecto HALL:	69
4. DISEÑO ELECTRÓNICO EN OrCAD:	71
<u>4.1. SOFTWARE OrCAD:</u>	<u>71</u>
4.1.1. OrCAD Capture CIS:	72
4.1.2. OrCAD PSpice:	72
4.1.3. OrCAD PCB Editor (Layout):	73
<u>4.2. DISEÑO DE ESQUEMAS ELECTRÓNICOS:</u>	<u>74</u>
5. DISEÑO SOFTWARE:	83
<u>5.1. PROGRAMACIÓN DEL PIC:</u>	<u>83</u>
5.1.1. Main.c:	84
5.1.2. Initialize.c:	84
5.1.3. Initialize.h:	84
5.1.4. Balancing.c:	84
5.1.5. Balancing.h:	84
5.1.6. Auxiliar.h:	85
5.1.7. Readsymux.c:	85
5.1.8. Readsymux.h:	85
5.1.9. Communications.c:	85
5.1.10. Thermos.c:	85
5.1.11. Thermos.h:	85
5.1.12. ADC.c:	86
5.1.13. ADC.h:	86
5.1.14. I2C.c:	86
5.1.15. I2C.h:	86
5.1.16. Structs.h:	86
<u>5.2. MÁQUINA DE ESTADOS PRINCIPAL:</u>	<u>87</u>
<u>5.3. DESIGNACIÓN DE ENTRADAS Y SALIDAS:</u>	<u>91</u>
<u>5.4. FUNCIONES PRINCIPALES:</u>	<u>95</u>

5.4.1. Funciones relacionadas con la inicialización:	95
5.4.2. Funciones relacionadas con los termómetros:	98
5.4.3. Funciones relacionadas con la comunicación I2C:	99
5.4.4. Funciones relacionadas con la comunicación USB:	102
5.4.5. Funciones relacionadas con el balanceo:	103
5.4.6. Funciones relacionadas con la verificación:	105
5.4.7. Funciones relacionadas con la barra de LED's:	106
5.4.8. Funciones relacionadas con el convertidor ADC.	107
6. PRESUPUESTO:	108
<u>6.1. COMPONENTES ELECTRÓNICOS:</u>	<u>108</u>
<u>6.2. MATERIALES DE FABRICACIÓN:</u>	<u>109</u>
<u>6.3. HORAS DE DISEÑO:</u>	<u>109</u>
<u>6.4. PRECIO TOTAL DEL PROYECTO:</u>	<u>109</u>
7. CONCLUSIONES:	110
<u>7.1. TRABAJOS FUTUROS:</u>	<u>111</u>
PLIEGO DE CONDICIONES	112
1. INTRODUCCIÓN:	113
2. CONDICIONES GENERALES:	113
3. ESPECIFICACIONES TÉCNICAS:	114
<u>3.1. COMPONENTES ANALOG DEVICES:</u>	<u>114</u>
<u>3.2. COMPONENTES TEXAS INSTRUMENTS:</u>	<u>116</u>
<u>3.3. COMPONENTES COSEL:</u>	<u>117</u>
<u>3.4. COMPONENTES SILICON LABS:</u>	<u>118</u>
<u>3.5. COMPONENTES MAXIM INTEGRATED:</u>	<u>119</u>

3.6. COMPONENTES MICROCHIP:	120
3.7. COMPONENTES ALLEGRO MICROSYSTEMS:	123
3.8. COMPONENTES ON SEMICONDUCTOR:	124
3.9. COMPONENTES RECOM POWER:	125
3.10. SOFTWARE:	126
ANEXO 1: DOCUMENTACIÓN TÉCNICA	127
ANEXO 2: ESQUEMAS ELECTRÓNICOS	179
ANEXO 3: PROGRAMACIÓN DEL MICRO PIC	187
ANEXO 4: SIMULACIÓN DEL DISPOSITIVO	278
SIMULACIÓN NÚMERO 1:	279
SIMULACIÓN NÚMERO 2:	285
SIMULACIÓN NÚMERO 3:	289

Índice de figuras:

Figura 1: Celdas no recargables.	20
Figura 2: Celdas recargables.	20
Figura 3: Baterías de Plomo-Ácido.	21
Figura 4: Baterías de Níquel-Cadmio.	22
Figura 5: Baterías de Níquel-Hidruro metálico.	22
Figura 6: Celdas Ión-Litio.	23
Figura 7: Celdas LiFePO ₄ .	24
Figura 8: Zona de operación de las celdas liFePO ₄ .	29
Figura 9: Topologías de los BMS	33
Figura 10: Máquina de estado del BMS.	36
Figura 11: Estado de carga y descarga	37
Figura 12: Photo-MOSFET Toshiba TLP240GA	45
Figura 13: Aislador digital	47
Figura 14: Circuito de acondicionamiento	50
Figura 15: Estructura interna del microcontrolador.	53
Figura 16: Arquitectura tradicional.	54
Figura 17: Arquitectura harvard.	55
Figura 18: Sensor de temperatura.	58
Figura 19: Conexiones USB.	61
Figura 20: Conexión entre dispositivos maestro-esclavo.	62
Figura 21: Transmisión de un byte.	63
Figura 22: Conexión del Interruptor ON/OFF.	65
Figura 23: Comportamiento del MOSFET STP130N8F7.	68

Figura 24: Led parásito ultrarrápido.	68
Figura 25: Sensor Hall.	70
Figura 26: Logo software OrCAD.	71
Figura 27: Visor de proyectos de OrCAD.	74
Figura 28: Esquema de balanceo	76
Figura 29: Conexiones multiplexores.	77
Figura 30: Conexiones USB, sensores y led	78
Figura 31: Conexiones entre placas y ON/OFF	79
Figura 32: Conexiones del microcontrolador PIC	80
Figura 33: Conexiones de la placa de potencia	81
Figura 34: Circuito de acondicionamiento de señal	82
Figura 35: Estructuras de variables	86
Figura 36: Definición de las señales conectadas	95
Figura 37: Osciladores existentes en el PIC	96
Figura 38: Campos existentes en la variable battery.info	97
Figura 39: Valores de los campos de battery.info	97
Figura 30: Definición de los puertos de los termómetros	98
Figura 41: características del I2C	101
Figura 42: Función para leer bits en I2C.	101
Figura 43: Función para escribir bits en I2C.	101
Figura 44: Función de reconocimiento de bits en I2C.	101
Figura 45: Barra de LEDs respecto al nivel de batería.	106
Figura 46: Aisladores y amplificadores Analog Devices.	114
Figura 47: Componentes Texas Instruments	116

Figura 48: Aprobaciones de los certificados RoHS y REACH.	117
Figura 49: Convertidor DC/DC Cosel	117
Figura 50: Módulo de comunicaciones USB.	118
Figura 51: Sensor de temperatura.	119
Figura 52: Diagrama de bloques general de la MCU.	121
Figura 53: Componentes Allegro Microsystems.	123
Figura 54: Componentes ON Semiconductor	124
Figura 55: Alimentaciones Recom Power.	125

Índice de tablas:

Tabla 1: Tipos de BMS.	33
Tabla 2: Placas necesarias para configuraciones de baterías.	35
Tabla 3: MOSFET's adecuados para el proyecto	44
Tabla 4: Multiplexor y amplificador diferencial elegidos.	47
Tabla 5: Propiedades del aislador digital.	47
Tabla 6: Componentes del circuito de acondicionamiento.	50
Tabla 7: Propiedades del Microcontrolador PIC.	57
Tabla 8: Alimentación no aislada.	57
Tabla 9: Sensor de temperatura elegido.	59
Tabla 10: Convertidor USB elegido.	60
Tabla 11: Estados de los MOSFET.	67
Tabla 12: MOSFET's adecuados.	67
Tabla 13: Sensores Hall adecuados.	70
Tabla 14: Variable DCHG_CHG.	87
Tabla 15: Variable CHARGED.	89
Tabla 16: Diagrama de flujo de la rutina principal	90
Tabla 17: Definición de los pines del PIC24FJ256DA210.	94
Tabla 18 : Casos posibles de la función de respuesta.	100
Tabla 19: Estados del valor devuelto "fault".	105
Tabla 20: Pin- Planner del PIC24FJ256DA210.	122

MEMORIA

DESCRIPTIVA



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

1. INTRODUCCIÓN:

1.1. REDACCIÓN DEL PROYECTO:

El presente proyecto titulado “Diseño y programación de un sistema de gestión de baterías (BMS) para baterías LiFePO₄”, se ha realizado como Trabajo Final de Máster en Ingeniería Mecatrónica en la Universitat Politècnica de València.

Dicho proyecto ha sido realizado por Marcos Ahulló Carbonell, asistido por un tutor interno a la universidad, Vicente Fermín Casanova Calvo, profesor de la asignatura de Control Aplicado a Sistemas Mecatrónicos, y por un tutor externo, de la empresa Robotnik Automation S.L.L., Miguel Moreno Muñoz.

1.2. JUSTIFICACIÓN DEL PROYECTO:

En estos tiempos, en muchos ámbitos de la industria aumenta potencialmente el auge de los componentes electrónicos y eléctricos, y con ellos la necesidad de baterías eléctricas más óptimas. Todo esto, sumado a la proliferación de los vehículos eléctricos para reducir los efectos contaminantes y las plataforma móviles y robots colaborativos para reducir costes y tiempos de producción, ha originado un reto mundial para conseguir baterías con mejores prestaciones.

Actualmente, está demostrado que las baterías de Litio son las que mejores prestaciones ofrecen, por eso el uso de ellas en los dispositivos electrónicos más pequeños. Sin embargo, la degradación rápida y la sensibilidad a las altas temperaturas hacen de éstas unos sistemas inestables con riesgo de inflamación o incluso explosión. Para evitar estas situaciones, cada batería requiere de un dispositivo adicional de seguridad incluido en su mismo sistema. Este dispositivo de seguridad es el BMS (*Battery Management System*), destinado a controlar tanto el equilibrado de la tensión de todas las celdas conectadas como la carga y la descarga de éstas.

El mercado actual presenta ya muchos tipos de dispositivos de control de baterías (BMS). Sin embargo, la mayoría de estos están diseñados para un tipo de batería muy concreto y generalmente presentan la imposibilidad de acceder al ajuste o configuración de muchos de sus parámetros. También existen los BMS's diseñados de forma genérica para contemplar su funcionamiento en la mayoría de tipos de celdas, no obteniendo todo su rendimiento en ningún caso.

En este proyecto se desarrolla una alternativa a todos estos BMS's. Éste permite el control máximo sobre diferentes configuraciones de celdas: desde cuatro celdas, dos en cada bloque; hasta 64 celdas. Con este BMS se puede ensayar y probar cómo afectan las diferentes configuraciones para obtener un rendimiento óptimo en cada tipo de celda y sistema de baterías. Además y más importante, permitirá a los fabricantes de baterías eléctricas poder usar un mismo BMS para todo tipo de baterías, únicamente será requerido ajustar los parámetros necesarios y óptimos para cada tipo de celda y los límites de la batería en su conjunto.

1.3. OBJETIVOS DEL PROYECTO:

El objetivo final del proyecto es diseñar y programar un sistema de gestión de baterías, *Battery Management System* modular, adaptable y totalmente configurable para baterías LiFePO₄, gestionado por un microcontrolador PIC.

Dicho BMS será operativo para sistemas de baterías desde cuatro celdas a sesenta y cuatro, integrando para ello las placas esclavas de potencia o control pertinentes. Con esto, permitirá a los diseñadores de baterías eléctricas unificar el uso de un solo BMS para todo tipo de configuraciones de baterías LiFePO₄. Obteniendo el rendimiento máximo de cada uno gracias a la posibilidad de ajuste de la mayoría de los parámetros significativos del dispositivo.

Las celdas, compatibles con el dispositivo diseñado, tendrán una tensión mínima de 2,8V, y una máxima de 3,6V, para garantizar la permanencia de los sistemas de baterías en la “Zona de Operación Segura”, con una corriente continua de 60A y una corriente de pico de 600A.

La tensión mínima de funcionamiento del BMS será 11V, voltaje obtenido con cuatro celdas conectadas dos en cada bloque y descargadas al mínimo, mientras que la máxima será de 57,6V a plena carga.

Como medidas de seguridad, el dispositivo diagnosticará continuamente el estado de cada una de las celdas y garantizará que ninguna sobrepase el voltaje máximo ni el mínimo. Tendrá la posibilidad de llevar conectadas hasta cuatro sondas de temperatura posicionadas en diferentes ubicaciones para mantener todos los espacios controlados, desde la tarjeta de potencia hasta la temperatura ambiente. Se integrará un interruptor

“ON/OFF” que permitirá desconectar la batería en cualquier momento sin necesidad de desconectar los polos físicamente. Además, se implementarán tres indicadores luminosos, para visualizar rápidamente algunos de los estados del dispositivo, como serán el fallo, la carga y la descarga.

El dispositivo permitirá consultar en todo momento el estado de carga de cada celda y de la batería en general, el estado de balanceo y las temperaturas mediante una conexión USB y una interfaz I2C, sin embargo, en este proyecto, sólo se presentará la conexión USB, dejando para proyectos futuros la aplicación con interfaz I2C.

1.4. METODOLOGÍA A SEGUIR:

La filosofía seguida para llevar a cabo este proyecto empieza por el diseño y elección de todos los componentes adecuados para el diseño de este dispositivo de gestión de baterías LiFePO₄.

Antes de empezar con el diseño, se lleva a cabo un estudio previo de los diferentes tipos de baterías que existen en el mercado y algunas que todavía están en fase alfa, de desarrollo o solamente en proyecto futuro. Recopilando también las diferentes propiedades, ventajas e inconvenientes que presentan cada uno de los tipos, junto con las aplicaciones más adecuadas para éstas.

Seguidamente, se enumerarán algunos conceptos clave para el mejor entendimiento de la relación entre todos los parámetros y un conocimiento más profundo de las magnitudes y unidades usadas en este ámbito.

Una vez en el mundo de las baterías, se describirá más profundamente la estructura de las baterías LiFePO₄ y sus valores de “Zona de Operación Segura”, concretando también su problemática, desde la más frecuente a la más peligrosa, así como algunos remedios y soluciones adoptados.

A continuación, se valoran los requerimientos mínimos que tendrá el BMS diseñado en este proyecto. Como la capacidad de monitorizar la batería, obteniendo información periódica de cada uno de los puntos de extracción: sensor hall para la corriente, sondas para la temperatura, celdas para el equilibrado de los voltajes etc. La capacidad de proteger la batería, diseñando un circuito de potencia con el ajuste suficiente como para que se produzca un corte en la carga o en la descarga cuando las celdas tienden a

salirse fuera de la Zona de Operación Segura (SOA). La capacidad de estimar el estado de la batería, calculando el nivel de carga (SOC) y el estado de salud (SOH) a partir de todos los sensores. La capacidad de maximizar las prestaciones de la batería con un sistema de balanceo adecuado. Y la capacidad de extraer información mediante dispositivos externos con un protocolo USB.

El siguiente paso es enumerar y describir los distintos tipos de BMS que atienden al uso comercial en estos momentos. Una vez llevada a cabo la descripción, se procede a la elección del formato más adecuado para la placa de control y de potencia. También se explica brevemente qué distribuciones y qué cantidad de placas, tanto de potencia como de control, serán necesarias para cada tipo de configuración de sistemas de baterías, así como un breve resumen de sus características.

Una vez con el formato del dispositivo de gestión definido, se clasifican las diferentes partes del sistema, explicando las características específicas de cada una de las dos placas, tanto la de potencia como la de control.

El diagrama de estados del apartado siguiente explica la dinámica de conmutaciones que llevará a cabo el dispositivo en todos sus procesos. Es decir, determina cada una de las rutas y acciones que puede tomar el dispositivo en función de la información extraída. Siendo los dos estados principales el de encendido y el de apagado.

A partir del siguiente apartado, se explica la estructura hardware que va a presentar el dispositivo y se empieza a explicar cada uno de los diseños, cálculos y elecciones de componentes necesarios para cada parte de la circuitería electrónica del que va a estar basado.

Empezando por la tarjeta de control con el circuito de balanceo, cálculo y elección del MOSFET de balanceo, el circuito de medidas, cálculo y elección del amplificador diferencial y el multiplexor, distribución y conexionado de las celdas, el circuito de acondicionamiento de la señal, cálculo y elección de los componentes de acondicionamiento, el controlador, los sensores de temperatura, los medios de comunicación, diferentes conexiones auxiliares y el circuito de alimentaciones.

Para la tarjeta de potencia, el cálculo y elección del MOSFET de potencia, cálculo y elección del diodo parásito de protección, cálculo del aislador digital y del sensor de corriente efecto Hall.

Una vez con todos los componentes adecuados y definidos, se utiliza el software de diseño electrónico OrCAD para generar los esquemas electrónicos y crear todas las conexiones de comunicación y alimentación entre los dispositivos. Dicho proyecto consta de 7 esquemas en los que engloba tanto la placa de control como la de potencia, el apartado de comunicaciones y el apartado de control del microcontrolador PIC.

Terminados los esquemas electrónicos, se programa el microcontrolador PIC en lenguaje C con la ayuda del software MPLAB X IDE v5.4..

MPLAB es un entorno de desarrollo integrado (IDE) que permite disponer en el mismo entorno de varios compiladores a la vez, facilitando así la compilación de los diferentes proyectos. En este caso, se usa el compilador XC16 de Microchip apropiado para microcontroladores de 16 bits.

2. ESTUDIO PREVIO:

2.1. CONCEPTOS BÁSICOS SOBRE BATERÍAS:

Las baterías son dispositivos que consisten de una o más celdas electroquímicas, cuya función es almacenar la energía eléctrica proveniente de algún actuador externo. Cada celda se compone de un electrodo positivo, o cátodo, de un electrodo negativo, o ánodo, y de electrolitos que permiten que los iones se muevan entre los electrodos permitiendo que la corriente fluya fuera de la batería.

La clasificación más generalizada que pueden tener estas celdas es relativa a la capacidad o incapacidad de recarga.

- Celdas o baterías primarias: No permiten su recarga.
- Celdas o baterías secundarias: Sí permiten su recarga.

En este proyecto se utilizarán las de tipo secundario, debido a que el dispositivo a diseñar es, entre otras cosas definidas y explicadas más adelante, un estabilizador para mantener la carga y la descarga de estas celdas en la *Zona de Operación Segura*.



FIGURA 1: CELDAS NO RECARGABLES.



FIGURA 2: CELDAS RECARGABLES.

2.2. TIPOS DE BATERÍAS:

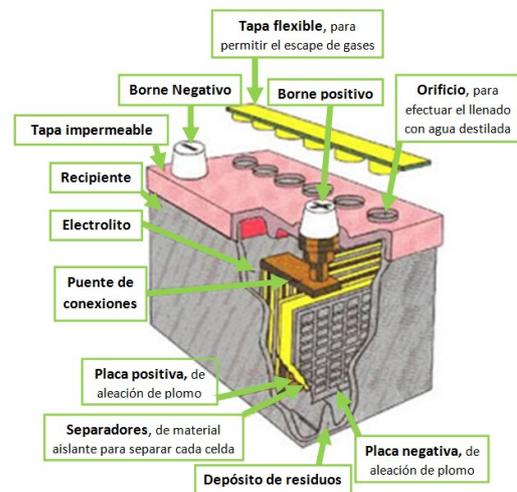
Hay muchos tipos de baterías diferentes. Dependiendo de la naturaleza interna y sus características electroquímicas tendrán unas propiedades u otras:

- Baterías de plomo-ácido:

Formadas por electrodos de plomo bañados en electrolitos de ácido sulfúrico. Debido a su asequibilidad y a la facilidad con la que se podían fabricar, fueron las principales baterías usadas en los primeros vehículos eléctricos existentes. Además de una vida bastante corta, alrededor de los tres años, tienen un impacto ambiental muy fuerte si no se reciclan. Se trata de una tecnología totalmente desfasada y superada en prestaciones hoy en día, además de inadecuada para el desarrollo del vehículo eléctrico por su baja densidad energética y pobre eficiencia (70%-75%).



FIGURA 3: BATERÍAS DE PLOMO-ÁCIDO.



- Baterías de Níquel-Cadmio (Ni-Cd):

Están formadas por electrodos de Cadmio bañados en un electrolito de Hidróxido de Potasio. Funcionan bien en un amplio rango de temperaturas y se pueden sobrecargar sin sufrir daños. Admiten descargas profundas y proporcionan un buen número de ciclos, pero acusan mucho el efecto memoria. Su peso y volumen, aunque mejores que los de las baterías de plomo-ácido, siguen siendo elevados para la energía que almacenan.

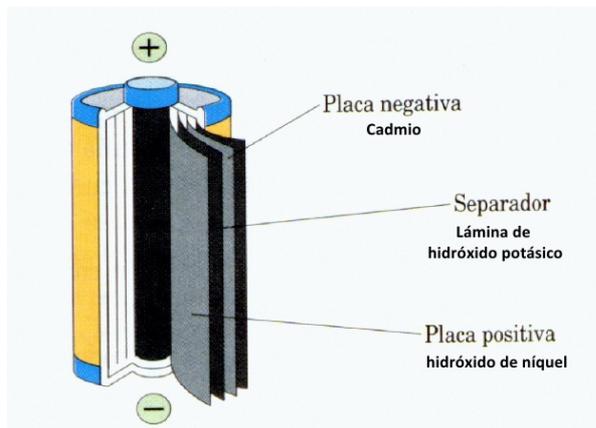


FIGURA 4: BATERÍAS DE NÍQUEL-CADMIO.

- Baterías de Níquel-Hidruro metálico (Ni-MH):

Utiliza un cátodo de Oxihidróxido de Níquel (NiOOH) y un ánodo de una aleación de Hidruro Metálico. Su densidad energética llega hasta los 100Wh/kg, superando las de plomo. Este tipo de baterías dispone de muchos ciclos de vida, siendo esto una ventaja para su aplicación en vehículos híbridos, pero con una baja eficiencia (60% - 70%) y un bajo rendimiento a temperaturas bajas. Por todo esto, no se usa hoy en día para vehículos eléctricos.



FIGURA 5: BATERÍAS DE NÍQUEL-HIDRURO METÁLICO.

- Baterías Ión-Litio (Li-Ion):

Es el tipo de batería más usado cuando se busca poco peso y volumen, por eso sus principales aplicaciones son ordenadores portátiles, tabletas, móviles, así como la mayoría de vehículos eléctricos actuales.

El esquema químico está formado por electrolitos de iones de litio, el ánodo de grafito y el cátodo de óxido de cobalto, trifilina u óxido de manganeso. Con esta estructura química interna, se es capaz de conformar baterías con una potencia y densidad energética de más de 250 Wh/kg en las últimas investigaciones, aunque los vehículos eléctricos utilizan versiones de menor densidad a cambio de conseguir una baja inflamabilidad, recarga rápida, mayor durabilidad y mayor respeto al medio ambiente. Así mismo, la eficiencia de la carga descarga es de un respetable 80-90%.

Esta clase de baterías serán las elegidas para su gestión, control y balanceo en este proyecto. Centrándose en la variedad de LiFePO_4 debido a su alto uso en el campo de la automoción, robótica y medicina.

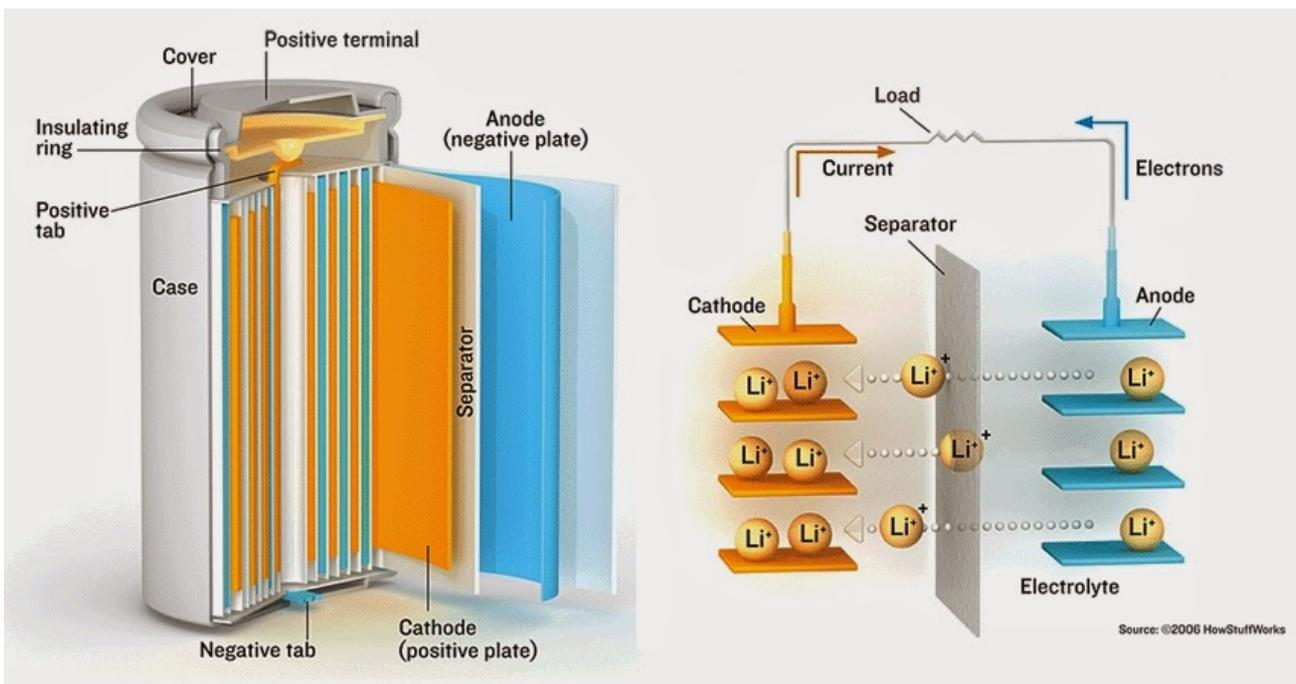


FIGURA 6: CELDAS IÓN-LITIO.

- Baterías de Polímero de Litio Ferroso (LiFePO₄):

La batería LiFePO₄ utiliza una química derivada de la tecnología de ión de litio, y comparte muchas ventajas de las baterías de este tipo. Sin embargo, hay diferencias sustanciales. La química LFP presenta una durabilidad mayor que otras baterías de ión de litio.

Al igual que las baterías recargables basadas en níquel, (diferenciándose de algunas baterías de iones de litio), las baterías LiFePO₄ tienen un voltaje de descarga muy constante. Su voltaje se mantiene cerca de los 3,2V durante la descarga hasta que la pila se vacía. Esto permite a la batería entregar virtualmente toda la energía hasta su descarga completa. Y puede simplificar mucho o incluso eliminar la necesidad de circuitería de regulación de voltaje. Excelentes características en seguridad, esto convierte a la LFP en una buena alternativa para las baterías de plomo en muchas aplicaciones como vehículos y solar, siempre que los sistemas de carga estén adaptados para no dañar las baterías LiFePO₄. La utilización de fosfatos evita el coste del cobalto y riesgos medioambientales.

El compuesto LiFePO₄ presenta corrientes o potencias instantáneas más altas que el LiCoO₂, además de tener una densidad de energía (energía/volumen) alrededor de un 30% más baja. Las celdas LiFePO₄ experimentan un ratio menor de pérdida de capacidad (más longevidad) que otras químicas como la LiCoO₂ (cobalto), LiMn₂O₄ (litio-manganeso) o LiPo (polímero de litio).



FIGURA 7: CELDAS LIFEPO4.

2.3. BATERÍAS EN FASE DE DESARROLLO:

Estas baterías citadas y explicadas anteriormente, son las que durante estos últimos años se han ido descubriendo mediante mucho tiempo de investigación. Además de todo esto, existen tipos de baterías en fase experimental, las cuales tomarán el relevo de las de Ión Litio en unos años. Las más desarrolladas actualmente son:

- Baterías con nanotecnologías:

Se podría catalogar como un tipo de ánodo que se aplica a las baterías actuales. Gracias a una microestructura especial, el ánodo multiplica la superficie de contacto con el electrolito mejorando espectacularmente las prestaciones de las baterías sin aumentar su peso ni su volumen.

- Baterías de litio en estado sólido:

El electrolito de litio es de metal en vez de líquido. Gracias a esta estructura, la densidad energética y la eficiencia de la batería aumentan y disminuye notablemente su inflamabilidad. También se mejoran los tiempos de carga y la vida útil de la batería.

- Supercondensadores:

El condensador es el acumulador eléctrico perfecto, es un almacén físico de electrones. Su velocidad de carga y descarga y su vida útil son casi ilimitadas y por tanto, incomparables con el resto de baterías. Su eficiencia es prácticamente del 100%. El problema es que aún no se ha descubierto cómo incrementar la densidad energética, una vez se descubra, podría ser la solución definitiva.

- Baterías de metal-aire:

Las hay de zinc y de aluminio. En lugar de un electrolito cerrado, la batería utiliza el oxígeno del aire para regenerarse, pero aún está en fase temprana de desarrollo.

2.4. PARÁMETROS CLAVE:

Se pueden definir cinco características básicas de las baterías:

- Densidad energética:

Cantidad de energía que es capaz de almacenar una batería en relación a su peso. Es el parámetro más importante a la hora de elegir una batería para cualquier aplicación, pues es el que más influye en autonomía y prestaciones. Se expresa en Wh/kg (Wattios-hora por kilogramo) lo cual quiere decir energía almacenada por cada kilogramo de peso de batería.

- Potencia específica:

Potencia que puede proporcionar cada kilogramo de peso de la batería, se expresa en W/kg. Cada batería tiene un límite característico de potencia a entregar, según cuál sean los electrodos y electrolitos tendrá más o menos potencia específica. No es recomendable sobrepasar el límite característico para el buen funcionamiento de la batería.

- Eficiencia de carga/descarga:

Es la relación entre la energía introducida a la batería en la fase de carga y la que realmente se almacena, las baterías modernas pueden llegar a tener una eficiencia del 90%, siendo el 10% restante disipado en forma de calor por las reacciones químicas que tienen lugar en el interior de cada celda.

- Ciclos de vida:

Las baterías recargables suelen perder capacidad a medida que aumenta el número de recargas. Depende de la tecnología y los compuestos de los que estén hechas, pueden ir desde varios cientos a varios miles de recargas.

- Velocidad de recarga:

Es el tiempo que se necesita para la carga de la batería. Suele medirse desde el inicio de la carga hasta el 80%.

2.5. PROBLEMÁTICA CON LAS BATERÍAS LiFePO₄:

El principal problema de las baterías de Litio es que están fabricadas con materiales altamente inflamables e inestables tanto en sus propiedades puras con la naturaleza, como formando parte de una aleación. Por ello, son necesarios los circuitos electrónicos adicionales para controlar el estado de la batería en todo momento, gestionando y controlando su carga y su descarga, además del balanceo de cada celda componente de la batería. Con estos dispositivos, se puede conseguir también que la descarga sea sensiblemente más rápida.

Estas baterías no resisten bien los cambios bruscos de temperatura ni son totalmente fiables en temperaturas extremas, tanto altas como bajas. La zona segura de operación puede estar entre -20°C y 60°C en descarga y de 0°C a 40°C en carga, todo ambiente con una temperatura fuera de estos márgenes, puede ocasionar problemas irreparables en las celdas de estas baterías. Además, estas baterías pueden sufrir envejecimiento al paso de los años, por tanto, cuanto más radical sea el ambiente en el que trabajen, más envejecimiento podrá presentar en menos tiempo.

A partir del conocimiento de estos problemas, para las baterías, se establece una zona de operación segura, llamada "SOA" (Safe Operating Area). Esta zona está delimitada por la temperatura, intensidad y voltaje. Éstas son las variables que el sistema de gestión de baterías, "BMS", va a tener que controlar en todo momento.

- Tensión de carga máxima: Toda celda tiene un límite máximo de voltaje, en el caso de las LiFePO₄, 3.6V, si se sobrepasa éste durante la fase de carga podría ocasionar quemaduras en la celda o incluso explosiones.

- Tensión de descarga máxima: Toda celda tiene un límite mínimo del voltaje, en el caso de las LiFePO_4 2.5V, si en algún momento el nivel de voltaje se quedara por debajo de éste, podría dañar las celdas e inhabilitar la batería.
- Temperatura: La vida útil de una celda se reduce drásticamente si se trabaja, tanto en la fase de carga como de descarga, en un ambiente con altas temperaturas.
- Corriente de carga y descarga máxima: Este tipo de celda tiene unos límites de intensidad, la vida de ésta se reduciría sensiblemente si la carga o la descarga se procesaran de manera muy rápida.

Estos límites pueden oscilar considerablemente dependiendo del fabricante de las baterías, o de su química. En la ilustración siguiente se presenta la zona de operación segura de las baterías LiFePO_4 .

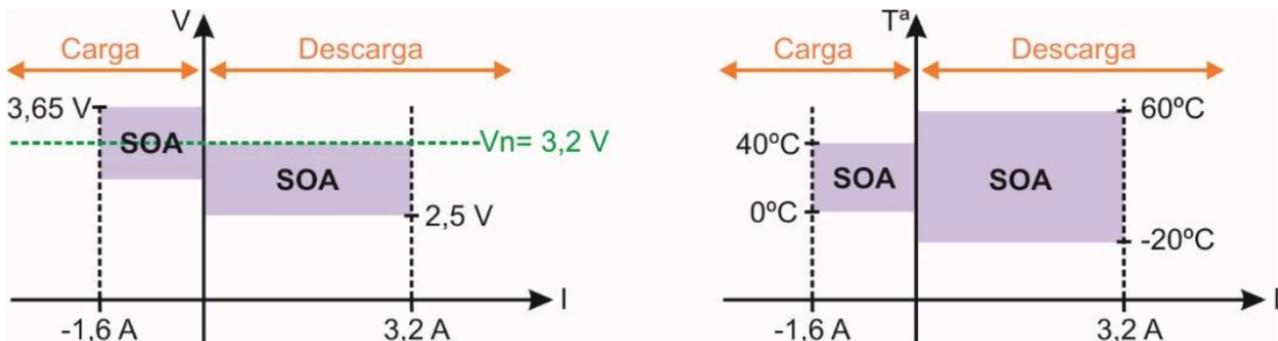


FIGURA 8: ZONA DE OPERACIÓN DE LAS CELDAS LIFEP04.

Para aplicaciones industriales y en general para la mayoría de dispositivos hay que conectar en serie o en paralelo varias celdas para poder llegar al voltaje y capacidad deseados. Por ello, se hace más complicado el sistema de carga, ya que en este tipo de baterías, es necesario que todas las celdas estén igualmente cargadas, para así garantizar una carga máxima de cada una de éstas y por tanto, del conjunto de batería.

Lo ideal es que el cargador reparta la tensión a partes iguales por todas las celdas, sin embargo, normalmente no es así debido a pequeñas tolerancias en la fabricación de las celdas, por tanto, en la etapa de carga, una celda podría llegar al tope de su capacidad sin haber llegado todavía las demás. En este caso, el BMS cortaría el proceso de carga para respetar la zona de operación segura de esa celda y no dejar opción a un mal tratamiento, explosión o reducción de su vida por alcanzar un voltaje más alto del límite.

Por todo esto y para garantizar la máxima capacidad de la batería, el BMS deberá emplear un método, llamado balanceo, para equilibrar los voltajes de todas las celdas entre sí, reduciendo la intensidad de carga en las celdas más cargadas y dejando al resto que se carguen hasta su máxima capacidad.

Este método estará controlado por el microcontrolador PIC, el cual mediante unas salidas digitales, activará los MOSFETs correspondientes a las celdas más cargadas para que dicha intensidad de carga pase por una resistencia conectada en paralelo, que será calculada en el apartado “3.4.2. Cálculo y elección de componentes de balanceo”.

2.6. FUNCIONES DEL BMS COMO SOLUCIÓN:

Una vez estudiada la problemática de las baterías Li-ion, y en concreto de las LiFePO₄, la solución es un dispositivo electrónico cuya composición y circuitería garantice que cada celda trabaje siempre en la zona de operación segura y equilibrada respecto a cualquier otra del sistema mediante un microcontrolador, un circuito de medidas y un circuito de balanceo entre otros aspectos. Por otro lado, las funciones que tendrá que llevar a cabo son:

- Maximizar las prestaciones de la batería:

Controlar el método de balanceo para garantizar la permanencia de los parámetros de las celdas dentro de los límites de la zona segura de operación y equilibrar todas las celdas para obtener su máxima capacidad y su máximo rendimiento.

- Monitorizar la batería:

Procesar toda la información de los sensores: voltajes, intensidades y temperaturas para incrementar el nivel de seguridad del dispositivo frente al sistema de baterías conectado.

- Estimar el estado de la batería:

Diagnosticar nivel máximo de carga (SoC) y nivel de salud (SoH) por medio de la información de los sensores y con el diseño de un algoritmo de cálculo mediante el PIC.

- Proteger la batería:

Garantizar que no se exceda el límite de voltaje en las celdas deteniendo la carga o la descarga, evitar que la temperatura del conjunto de la batería exceda el límite, deteniendo el sistema o enfriándolo con un disipador, impedir que la intensidad de carga y de la descarga sea mayor de lo que pueda soportar el circuito de potencia diseñado.

- Exportar todos los datos:

Permitir la publicación de toda la información por medio de una pantalla o una interfaz.

2.7. TIPOS DE BMS EXISTENTES:

Los BMS, por su precisión en las medidas y por su topología, característica importante ya que determina el coste, la facilidad de instalación y mantenimiento, podrían clasificarse en:

- Centralizado:

Es un BMS montado en la batería (o cercano a ella). Su correcta conexión se produce conectando un cable desde cada pin de entrada del BMS a cada celda existente en el grupo de batería. Tiene las ventajas de ser muy fácil y rápido de ensamblar, además de ser muy compacto.

- Modular:

Usa la misma tecnología que el centralizado, pero teniendo repartidos diferentes módulos por el grupo de baterías y siendo éstos los encargados de recoger y enviar toda la información obtenida de cada una de las celdas conectadas. Normalmente, se utiliza la tecnología “Maestro-Esclavo”, siendo el maestro el que controla todos los módulos y las comunicaciones entre ellos. Dado que los módulos están más cerca de las celdas a medir y los cables son sensiblemente más cortos, se puede reducir notablemente el ruido y la caída de tensión de las medidas y por tanto, aumentar la precisión de las mismas.

- Maestro-Esclavo:

Usa una tecnología muy similar al modular, presenta varios módulos esclavos que realizan medidas sobre varias celdas cada uno y después envían la información al maestro para su gestión. Las ventajas de fabricación y coste son las mismas que con la tecnología modular, a excepción de que las placas esclavo son más baratas debido a la presencia de menos componentes de control que con el maestro.

- Distribuido:

Se conecta un pequeño circuito de medida a cada celda de la batería, obteniendo una medida lo más precisa posible y sin ruido. Posteriormente, se envían los datos de cada una por un cable de comunicaciones conectado a un circuito controlador y se procesan y gestionan. Al tener más material que implementar, es el tipo más caro de BMS, aunque también el más preciso.

Topología	Calidad de las medidas	Ruido	Versatilidad	Seguridad	Coste	Montaje	Mantenimiento
Centralizado	Bueno	Excelente	Regular	Regular	Regular	Bueno	Regular
Modular	Bueno	Excelente	Bueno	Bueno	Excelente	Excelente	Regular
Ma - Esc	Bueno	Excelente	Bueno	Bueno	Excelente	Excelente	Regular
Distribuido	Excelente	Bueno	Excelente	Excelente	Excelente	Regular	Bueno

Tabla 1: Tipos de BMS.

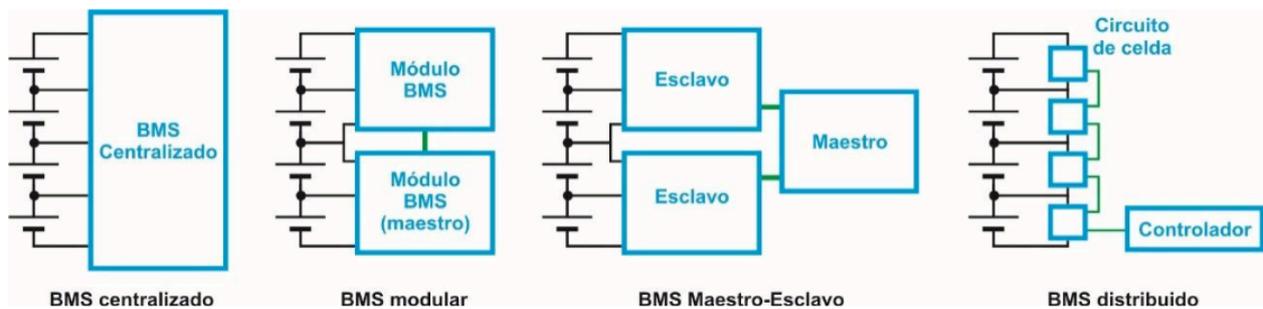


FIGURA 9: TOPOLOGÍAS DE LOS BMS

2.8. ELECCIÓN DEL TIPO DE BMS PARA EL PROYECTO:

Una vez comprobadas todas las tecnologías de los diferentes tipos de Sistemas de Gestión de Baterías (BMS) y comparadas las cuatro topologías enumeradas y explicadas anteriormente, se puede observar que la topología “Centralizada” es la que menores prestaciones ofrece. Sólo siendo valiosa su inmunidad al ruido. La mejor, por el contrario, es la topología “Distribuida”, ya que aunque es la tecnología más cara de las tres, ofrece unas prestaciones muy buenas y muy versátiles.

El coste de la topología “Distribuida” es más alto debido a que cada celda tiene su pequeño circuito de control de carga y, por tanto, hay que diseñar, construir y ensamblar más placas y componentes, gracias a la poca distancia entre el circuito de control y la fuente de voltaje, la celda, se pueden obtener unas mediciones con mucha más resolución que en cualquier otra topología. Por el contrario, el hecho de que el circuito de control y los cables de conexión estén inmersos en la electrónica de potencia pueden producir más ruido y con ello, fallos de medición o comunicación.

La topología “Modular” tiene mejores prestaciones, en general, que la “Centralizada”. Más versatilidad, más seguridad y menor coste. Con esta tecnología se puede depender del número de celdas que presente la batería para adherirle los módulos que sean necesarios y óptimos para llevar a cabo una correcta gestión, ahorrando así en coste de componentes, montaje y mantenimiento.

Por tanto, para el proyecto se ha elegido una topología Modular con un protocolo de comunicaciones de Maestro - Esclavo. Con esta tecnología se podrá adaptar el número de módulos que será necesario adherir al sistema para su correcta medición en base a

las celdas conectadas tanto en serie como en paralelo en el pack de baterías en cuestión, optimizando los costes para cada uno de los casos.

Se diseñará una placa maestra con el circuito de potencia y un circuito de medición y balance adaptado a 16 celdas. Todas las placas esclavo presentarán un circuito de medición y balance propio para conectar 16 celdas más. En cada sistema y por cada línea de celdas conectadas en paralelo, habrá una placa esclavo que contendrá, además, un circuito de potencia.

Con la posibilidad de ser modular, este dispositivo podrá abarcar desde una batería de cuatro celdas en serie (4S), hasta una de treinta y dos celdas por dos filas en paralelo (32S2P). Para obtener todas estas prestaciones, es necesario diseñar un BMS a medida usando electrónica analógica.

La siguiente tabla describe las placas que serán necesarios para controlar diferentes tipos de sistemas de baterías.

	8S1P	8S2P	16S1P	16S2P	32S1P	32S2P
Placa Maestra	1	1	1	1	1	1
Placa Esclava con potencia y comunicación	0	0	0	1	0	1
Placa Esclava con comunicación	0	0	0	0	1	2
Total	1	1	1	2	2	4

Tabla 2: Placas necesarias para configuraciones de baterías.

Eligiendo la solución de gestionar dieciséis celdas por cada tarjeta, se cubre el rango más amplio de aplicaciones de plataformas móviles. De este modo, la mayor parte de los sistemas de baterías producidos en la empresa estarán gestionados por una sola placa.

3. DISEÑO HARDWARE:

3.1. DIAGRAMA DE ESTADOS DEL BMS:

El funcionamiento, como en la mayoría de dispositivos está regido por un diagrama de estados, el cual controla todas las rutas y acciones que puede llevar a cabo dependiendo de la información que se haya recogido y el sistema de baterías conectado.

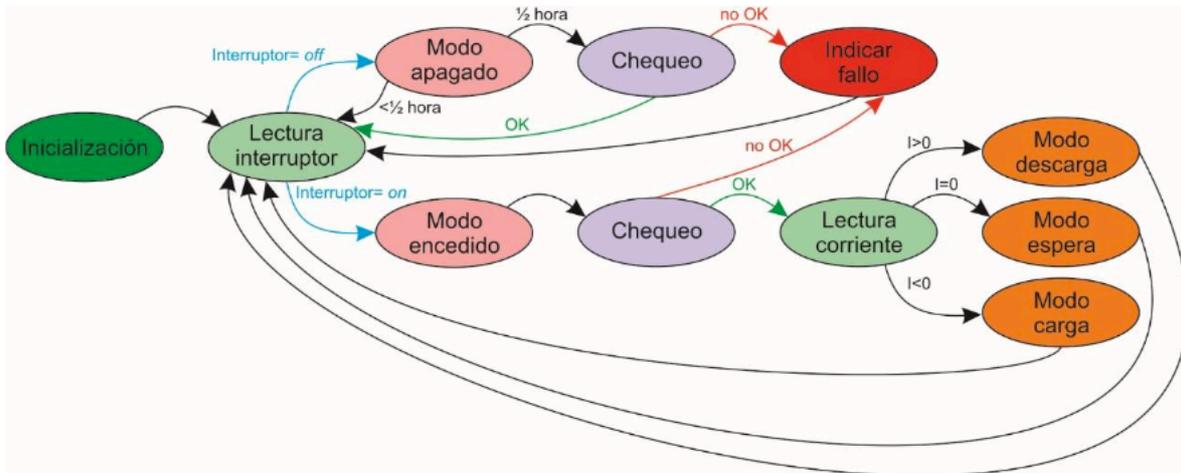


FIGURA 10: MÁQUINA DE ESTADO DEL BMS.

Los estados generales del BMS son “OFF” y “ON”.

Estado OFF:

Se produce una reducción de las funciones del BMS para ahorrar la máxima energía posible en la auto-descarga del sistema de baterías. En este modo, se produce una medición de los parámetros básicos de seguridad (tensión de celdas, temperatura de la batería etc.) cada media hora aproximadamente. Si alguno de los parámetros resultara fuera de rango, se encendería el LED de fallo de la placa para avisar al usuario.

Estado ON:

Una vez en el estado de encendido, se podrán alcanzar tres estados en la misma máquina. El estado de espera, que estará activado cuando el dispositivo no tenga ningún sistema de baterías conectado. El estado de carga, cuando haya una fuente de alimentación conectada entre los bornes. Y el estado de descarga, cuando haya cualquier tipo de carga conectada entre los bornes.

Para que se den estos casos, el BMS está configurado por defecto para mantener disponible la circulación de corriente por ambos sentidos, si el medidor de corriente obtiene cero, significa que no hay ningún sistema de baterías conectado y por tanto estará en el estado de espera. Si la lectura que obtiene el medidor de corriente es positiva, el dispositivo estará en el estado de descarga, ya que habrá una carga consumiendo corriente de la batería. Si la medición de corriente es negativa, el dispositivo estará en el estado de carga, ya que una alimentación habrá sido colocada entre los bornes de la batería.

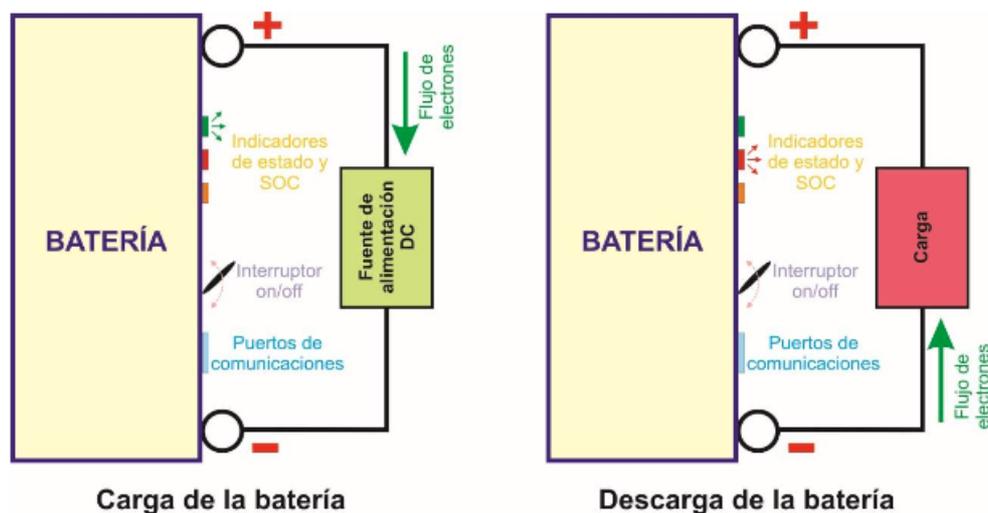


FIGURA 11: ESTADO DE CARGA Y DESCARGA

3.2. PARTES DEL BMS:

El BMS vendrá formado por dos tipos de tarjetas, la tarjeta de potencia y la tarjeta de control.

Dependiendo del número de celdas presentes en el sistema de baterías a gestionar, este dispositivo se montará con las placas pertinentes para el control de todas las celdas en cuestión.

Se necesitará una placa de potencia para cada rama en paralelo del sistema, siendo la primera siempre la maestra y a partir de la segunda línea en paralelo, placas esclavas; y una placa de control por cada dieciséis celdas conectadas en serie.

La tarjeta de potencia es la encargada de controlar la corriente de carga y descarga del sistema de baterías mediante los dispositivos electrónicos de potencia. También es la responsable de la medida de la corriente instantánea que está circulando, y con esta información recopilada, se convierte en el dispositivo emisor de la comunicación con la tarjeta de control para que se lleve a cabo un procesamiento de la misma, y por tanto, una buena gestión del sistema de baterías. Ésta misma, mediante LED's, indica el estado en el que se encuentra la batería.

A la tarjeta de control se conecta el voltaje de control de cada una de las celdas, con un máximo de dieciséis por tarjeta. Ésta es la encargada de realizar medidas de voltaje de todas las celdas conectadas y de realizar el proceso de balanceo en la etapa de carga del sistema, en el caso de que entre una celda y otra haya una diferencia de potencial más alta que 3,6V o que tenga una diferencia de potencial más grande que un 0,7% del voltaje de la celda medida, el MOSFET de control se activará y permitirá la disipación de

la corriente necesaria por medio de la resistencia conectada en serie al MOSFET y en paralelo a la celda. También se encarga de establecer las comunicaciones USB entre ella misma y la tarjeta de potencia.

El microcontrolador PIC (Programmable Integrated Circuited) es el encargado de la gestión y el control completo de la tarjeta maestra de control, además de efectuar comunicación y coordinarse con el resto de tarjetas. Al disponer puertos dedicados para ello, la comunicación I2C se gestiona directamente desde el PIC, al igual que gestiona el botón de encendido y apagado general del dispositivo y del sistema. Además, gestiona la apertura y cierre de los dispositivos que permiten e imposibilitan se efectúe la carga o la descarga sobre el sistema de baterías.

3.3. REQUISITOS DEL BMS:

El BMS está compuesto, en todos los casos, por una placa de potencia y otra de control. La de potencia es la encargada de abrir y cerrar los contactos y los dispositivos que permiten la carga y la descarga de la batería.

La tarjeta de control, es la placa encargada de la medición y el procesamiento de los datos de todo el sistema.

El apartado de potencia siempre tendrá que ser más robusto que el de control, dado que por los contactos de potencia es necesario que pase toda la intensidad demandada por la carga conectada y toda la intensidad ofrecida por el cargador del sistema de baterías.

En el caso del control, la apertura y cierre de dispositivos y bobinas, la realización de mediciones etc. mediante señales, emplea mucha menos intensidad en cada conexión y por tanto, es necesaria menos robustez.

Como el BMS se alimenta de la propia batería que gestiona y el voltaje varía en función del estado de carga en el que se encuentre, se seleccionarán los componentes teniendo en cuenta las características siguientes:

- Mayor rendimiento.
- Menor consumo.
- Mayor margen de alimentación.
- Mayor inmunidad a las variaciones de alimentación.

3.4. TARJETA DE CONTROL:

En esta tarjeta estarán presentes los circuitos de medición y balanceo de las celdas, y una parte de éstas se deben alimentar directamente del voltaje del bloque de celdas, por tanto, van a tener que poder adaptarse a un voltaje con una variación muy alta, porque éste va a depender del número de celdas conectadas, del tipo de celdas conectadas y del estado en el que se encuentren (voltaje a plena carga, voltaje nominal o voltaje mínimo).

Por el contrario, el circuito de comunicaciones y el controlador necesitarán un voltaje estable y fijo, logrando esto con la conexión de un convertidor dc/dc aislado de 3.3V en la placa directamente a batería, de este modo se consigue un consumo por igual en todas las celdas y se favorece al balanceo del propio sistema.

Es recomendable conectar a la placa un convertidor dc/dc con el mayor rango de voltaje de entrada, porque éste fijará el número mínimo y máximo de celdas que se deberán conectar para que el dispositivo funcione.

Los circuitos de acondicionamiento y de aislamiento son los encargados de adaptar y aislar estos dos entornos, el variable para los circuitos de medida y balanceo, y el fijo para los controladores y el circuito de comunicaciones.

3.4.1. Circuito de balanceo:

Este circuito controla el equilibrio de todas las celdas en el sistema de baterías, es decir, adapta la carga y la descarga para que todas las celdas presenten el mismo voltaje y por tanto, obtener el máximo rendimiento de ellas.

Hay dos sistemas diferentes de balanceo: la derivación por corriente y la redistribución de la carga.

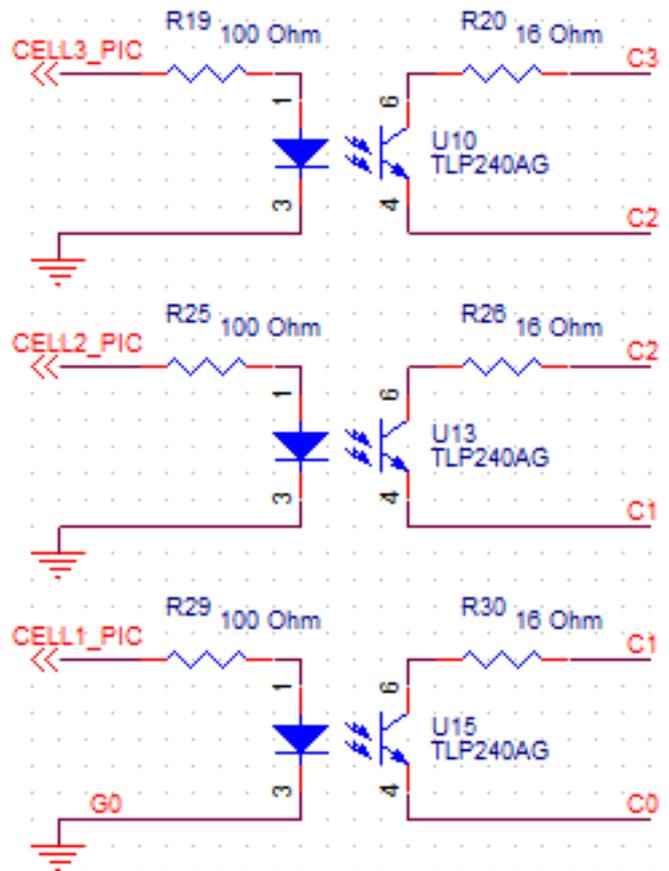
- La derivación de corriente consiste en reducir la velocidad de carga de las celdas con mayor tensión, desviando esta corriente excedente a través de una resistencia, la cual disipa toda esa energía con calor.
- La redistribución de carga se lleva a cabo del mismo modo, pero la corriente excedente se utiliza también para cargar las celdas con menos tensión.

En este proyecto se implementará la tecnología de la derivación de corriente, porque, aunque se desperdicie parte de la energía, su implementación e integración en la placa es más sencilla y es más apropiado para sistemas que tienen carga desde la red eléctrica.

La integración de este circuito se realiza conectando un MOSFET en paralelo por cada celda con una resistencia en serie, para que, cuando la diferencia de potencial, calculada por el algoritmo del controlador, pase de un determinado porcentaje (0.7%) respecto a la medida con menos tensión, el MOSFET se active y redirija la tensión de carga a dicha resistencia.

Este cálculo de diferencia de potencial puede llevarse a cabo periódicamente, o solamente al final de la última fase de carga.

El BMS del proyecto lo calcula periódicamente durante todo el proceso de carga, dado que la confección del controlador es relativamente más sencilla y permite tener un mayor conocimiento y control del estado de la batería.



3.4.2. Cálculo y elección del MOSFET de balanceo:

Para lograr un balanceo adecuado en los sistemas de baterías de las plataformas móviles diseñadas, el sistema tiene que presentar una corriente de balanceo de al menos 100mA. Para este proyecto se elige un MOSFET aislado que pueda ser controlado por señales digitales y se diseña su circuito para que su corriente de balanceo sea de 500mA. Este transistor, tiene como finalidad, conmutar señales electrónicas y controlar la descarga de cada celda.

La tensión máxima a la que puede llegar una celda LiFePO₄ cuando está en la última fase de carga, es decir, cargada al 100%, es de 3.6V, por tanto, esta es la tensión máxima que deben soportar los MOSFET's.

Como la corriente de balanceo máxima será de 0.5A, 500mA, se puede calcular la resistencia equivalente necesaria (siendo la suma de la resistencia de balanceo y la resistencia en conducción) en su circuito para que descargue a esa intensidad en concreto.

$$R_b + R_{on} = 3.6V/0.5A = 7.2 \Omega$$

Una vez calculada la resistencia equivalente, se buscará una resistencia de balanceo que, junto con la resistencia interna en conducción del MOSFET dé como resultado 7,2 Ω .

Esta resistencia se conecta en serie con el MOSFET elegido a continuación, y estos dos en paralelo con la celda correspondiente; este proceso se repetiría en cada una de las dieciséis conexiones a celdas presentes en la tarjeta de control del dispositivo.

MOSFET	V _f	I _f	R _{on}	I _{on} (máx)	R _b
TLP240GA	1,27V	10mA	2 Ω	500mA	5,2 Ω
TLP220A	1,63V	5mA	0,6 Ω	100mA	17,4 Ω

Tabla 3: MOSFET's adecuados para el proyecto

Debido a la voluntad de tener un índice de disipación más alto que 100mA para garantizar el balanceo de las celdas aunque la potencia de carga sea máxima, se elige el Toshiba TLP240GA por su alta corriente máxima en conducción y una nula diferencia en el precio con respecto al modelo TLP220A.

Al tener este MOSFET una resistencia interna en conducción de 2 Ω , será suficiente con establecer el valor de la resistencia de balanceo como 5,1 Ω ; valor normalizado y fácil de encontrar en cualquier tienda o plataforma web especializada.

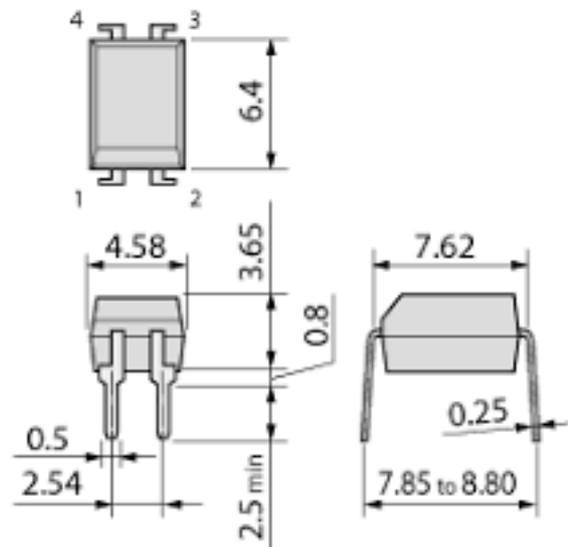


FIGURA 12: PHOTO-MOSFET TOSHIBA TLP240GA

-
- V_f o tensión delantera: Es la tensión nominal a la que se activa la puerta del MOSFET por medio de una salida digital.
 - I_f o corriente delantera: Es la corriente que necesita tener la tensión anterior para que se efectúe la activación de la puerta del MOSFET.
 - R_{on} o resistencia en conducción: Es la resistencia que opone el photo-MOSFET cuando la puerta se activa y entra en fase de conducción.
 - $I_{on(m\acute{a}x)}$ o corriente máxima en conducción: Es la corriente máxima que puede circular entre fuente y drenador, por tanto, este valor será el máximo al que se podría ajustar el balanceo de este modelo MOSFET.
 - R_b o resistencia de balanceo: Es la resistencia añadida en serie al optoacoplador para poder determinar la intensidad de disipación en el balanceo. La resistencia equivalente a tener en cuenta para calcula el valor es $R_{on} + R_b$.

3.4.3. Circuito de medidas:

La tarjeta de control programada para este dispositivo tiene que lograr medir dieciséis voltajes diferentes en modo diferencial, para obtener la medida individual de cada celda.

Para no ocasionar problemas en la medida de estos voltajes, se desestima la opción de integrar un conversor analógico-digital (ADC) de dieciséis canales diferentes. En lugar de esto, se opta por multiplexar la lectura de las celdas de cada bloque y usar un amplificador diferencial por bloque (cada ocho celdas). La medida de cada bloque se conectará a una entrada del microcontrolador PIC, en concreto a un canal comparador del ADC, a través de un circuito de acondicionamiento.

3.4.4. Multiplexor y amplificador diferencial:

Cada bloque tendrá dos tomas de alimentación: negativo y positivo. En el caso del bloque A, se indican como los puntos G0 y G1. Siendo G0 el negativo y G1 el positivo.

En el caso del bloque B, los puntos son G1 como negativo y G2, como positivo. (Pueden

Componente	Fabricante	Modelo	Alimentación	Nº Canales
Aislador Digital	Analog Devices	ADUM1400BRWZ	2,7V a 5,5V	4

verse dichas referencias en el Anexo 2: Esquemas electrónicos).

Estas alimentaciones distan mucho de ser constantes porque pasará tanto voltaje como celdas haya conectadas a la tarjeta de control. Sabiendo que el dispositivo de gestión puede actuar correctamente estando conectada una batería de cuatro celdas, o una de dieciséis (los dos bloques completos), se necesita un margen de alimentación muy amplio. La tensión máxima por bloque es de 28.8V, y la mínima es 5,6V con dos celdas descargadas. Entonces, se necesita que los dos componentes puedan ser alimentados en todo este rango de voltajes.

Componente	Fabricante	Modelo	Alimentación	Ganancia en BW
Multiplexor	Analog Devices	DG409DJ+	5V a 30V	550kHz
Amplificador Diferencial	Vishay	AD8276ARMZ-R7	2V a 36V	1MHz

Tabla 4: Multiplexor y amplificador diferencial elegidos.

En este BMS, las señales de control de los multiplexores son comunes. Cada vez que se efectúa una lectura de las señales de control, se selecciona la misma señal para ambos multiplexores y se habilitan con la misma señal selectora de control. También se necesita aislar galvánicamente para separarlas del circuito controlador.

El aislador digital adecuado para el proyecto es el ADUM1400BRWZ del fabricante Analog Devices, éste aísla las tres señales de selección provenientes del microcontrolador y la señal de habilitación del multiplexor analógico.

Tabla 5: Propiedades del aislador digital.

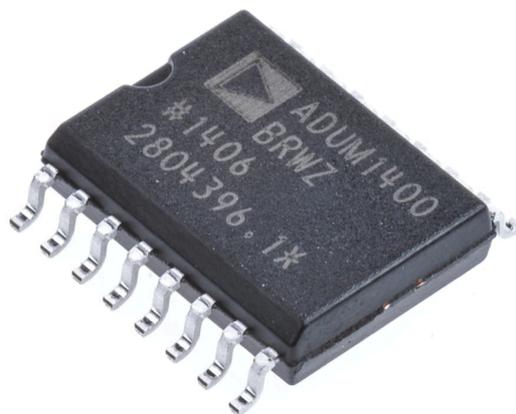


FIGURA 13: AISLADOR DIGITAL

3.4.5. Configuración de las celdas:

El límite máximo de celdas que se pueden conectar en un solo módulo de este sistema de gestión de baterías son dieciséis, ocho en cada bloque.

En el caso de que se quieran conectar menos de dieciséis, el límite mínimo de celdas para que pueda funcionar el dispositivo es cuatro, conectadas dos en cada bloque para que haya una diferencia de potencial de al menos 5V en cada circuito de medida. Todo el resto de pines de control tendrán que ser cortocircuitados con un puente hacia la zona de más potencial del bloque (G1 para el bloque 1 y G2 para el bloque 2).

Los puentes también son útiles para determinar en el controlador del dispositivo, qué celdas son las que realmente están conectadas y cuales no, dado que la tensión de salida del amplificador diferencial entre dos celdas no conectadas, sería 0V. Sin embargo, este método no garantiza la seguridad del dispositivo porque si una celda se degrada y ofrece 0V a su salida, el amplificador diferencial entenderá que en ese pin no hay conectada ninguna celda y en vez de reconocerlo como un “fallo”, reconozca “celda no conectada”.

Por ello, el usuario deberá indicar al sistema cuántas celdas hay conectadas y éste verificará si es correcto.

3.4.6. Circuito de acondicionamiento de la señal:

Una vez se obtienen las señales de salida de los amplificadores diferenciales de los dos bloques, hay que convertir éstas a digital para que puedan ser procesadas por el controlador. Sin embargo, estas dos señales están referenciadas a distintos puntos en el circuito del dispositivo, siendo el punto G0 para el bloque A y el G1 para el B. Como hay que referenciar las señales a la masa del circuito del controlador para que pueda reconocerlas, se utiliza un amplificador de aislamiento, que tiene como función, aislar dos circuitos entre sí permitiendo el paso de información entre ellos.

Por tanto, la parte no aislada de cada amplificador quedará alimentada con la tensión correspondiente al bloque de celdas del que recibe la señal, y la parte aislada, quedará alimentada con la misma tensión y referencia usadas para alimentar el controlador.

La señal del amplificador va a variar, en condiciones normales del sistema de baterías, entre 3V y 3,6V, que corresponde a las tensiones que abarca la zona segura de operación de cada una de las celdas. Sin embargo, el rango de entrada de este amplificador de aislamiento es de 0V a 2,46V. Por tanto, habrá que diseñar un pequeño circuito de acondicionamiento para que la señal de salida del amplificador diferencial de cada bloque se adapte al rango de entrada de este dispositivo.

3.4.7. Elección de los componentes de acondicionamiento:

Este es un circuito compuesto por elementos lógicos y resistencias que adapta cualquier señal al rango deseado.

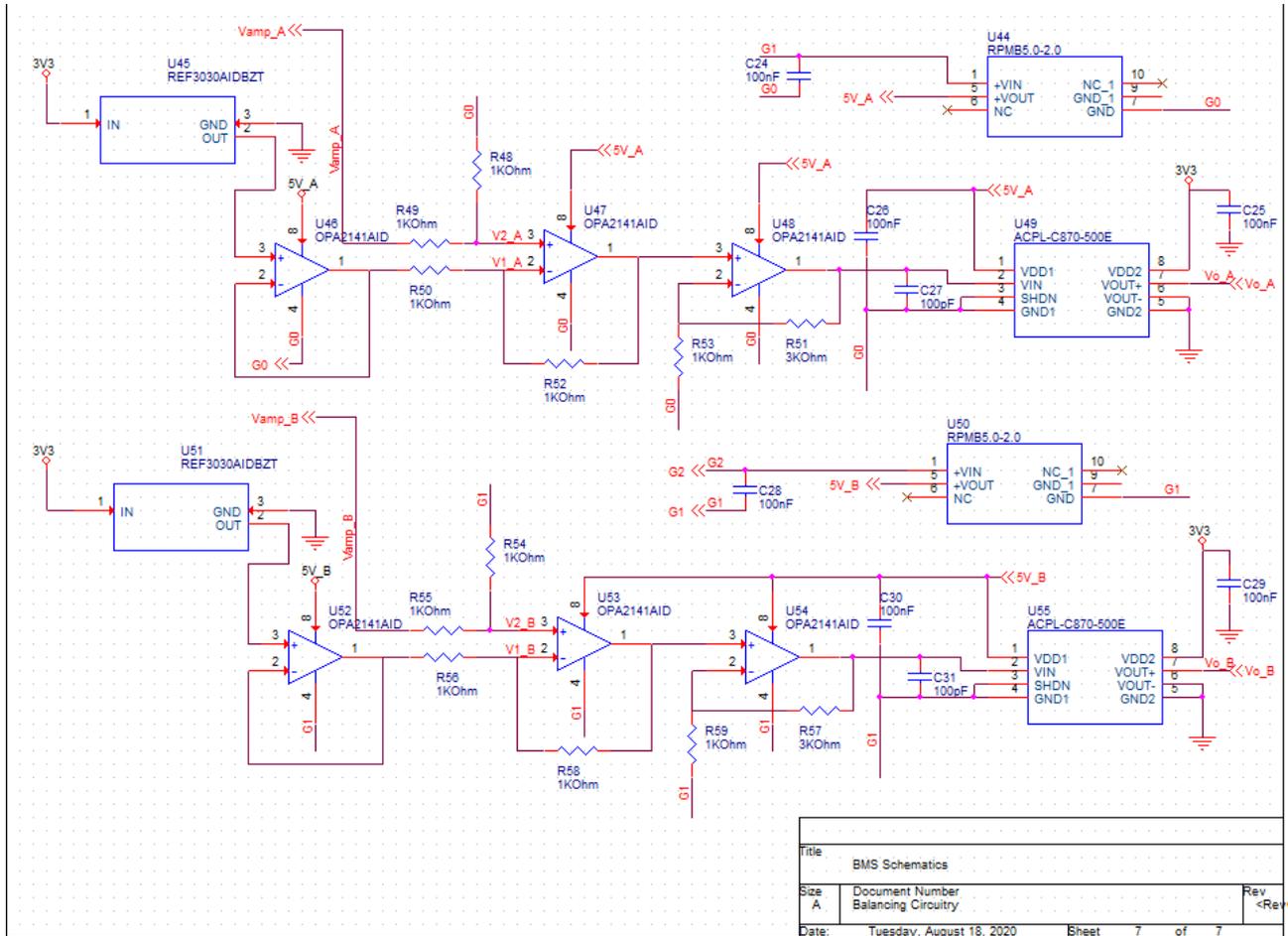


FIGURA 14: CIRCUITO DE ACONDICIONAMIENTO

Los componentes escogidos para el diseño de este circuito operacional son los siguientes:

Componente	Referencia	Fabricante	Modelo	Cantidad
Referencia de Tensión	A1	Texas Instruments	REF3330AIDBZT	1
Amplificador Operacional	AO1, AO2, AO3	Texas Instruments	OPA2141AID	3
Amplificador de Aislamiento	AO4	Allegro Systems	ACPL-C87A-500E	1
Convertidor DC/DC	A2	Recom Power	RPMB5.0-2.0	1

Tabla 6: Componentes del circuito de acondicionamiento.

Para convertir la señal de rango de voltajes de 2,8V - 3,6V a la señal de 0V - 2,46V, el primer paso es eliminar el offset de 2.8V, y el segundo paso es amplificar la señal de rango alto para ajustarse lo máximo posible al rango de entrada del amplificador de aislamiento y así favorecer la precisión de la medida.

Para eliminar dicho offset, se obtiene un valor constante de 3V desde la referencia de tensión, que, seguida de un amplificador buffer, o seguidor de tensión (AO1), eleva el rango mínimo de voltaje de la entrada negativa al restador (AO2) a 3V, resultando dicha tensión fija en V1 y siendo V2, la entrada positiva, el rango de voltajes obtenido en las medidas de las celdas, entre 3V y 3,6V, y por tanto, reduciendo la diferencia de voltaje entre la entrada negativa y positiva del restador, se obtienen 0,6V.

Este seguidor de tensión es un amplificador operacional que tiene una ganancia de voltaje 1. Esto significa que no proporciona ninguna amplificación ni atenuación de la señal, su propósito principal es reducir el consumo de corriente de su tensión por tener una alta impedancia en la entrada, y así, evitar perturbaciones y ruidos perjudiciales que afectan a la estabilidad de las señales medidas.

Una vez obtenida esta tensión sin ruido, y en paralelo con la señal de salida del amplificador diferencial del circuito de medida de las celdas, se conecta el restador (AO2), cuya función de transferencia es:

$$V_{out} = \frac{R_2}{R_1} (V_2 - V_1)$$

Ya que el propósito de este amplificador operacional es realizar una diferencia entre las dos señales entrantes sin producir ninguna amplificación ni atenuación, las resistencias R1 y R2 deberán ser iguales. Se escoge 1kΩ como valor de compromiso, por no ser

demasiado elevado para añadir mucho ruido, ni demasiado bajo para que las corrientes sean elevadas.

A la salida de este restador, se obtiene un rango de voltajes entre 0V y 0,6V. Una vez con el límite mínimo ajustado, sólo queda ajustar el límite máximo, que tiene que llegar a los 2,46V.

Para ello, se diseña un amplificador operacional no inversor con ganancia 4, cuya entrada positiva será la señal filtrada anteriormente con el restador, y la señal negativa será la referencia a la que está conectada cada bloque de celdas, en el caso del circuito del bloque A de celdas, G0 y en el del bloque B, G1.

La función de transferencia de este amplificador no inversor es:

$$V_{out} = \left(1 + \frac{R_2}{R_1}\right) V_{in}$$

Con esta función se extrae que, para que la ganancia del amplificador sea 4, la relación entre R3 y R4 tiene que ser 3. Por tanto, se vuelve a escoger 1kΩ como valor de compromiso para la resistencia R3 y por consiguiente, R4 deberá valer 3kΩ.

Operando con su función de transferencia, el voltaje máximo correspondiente a la salida del amplificador no inversor es de 2,4V, adaptándose de forma muy precisa al rango demandado por el amplificador de aislamiento que enviará esa señal al conversor ADC del microcontrolador para ser procesada de forma adecuada por el controlador.

3.4.8. El microcontrolador:

El control de este sistema de gestión de baterías va a llevarse a cabo mediante un microcontrolador PIC (*Peripheral Interface Controller*).

Un PIC es un circuito integrado compuesto por una Unidad Central de Procesamiento (CPU), una unidad de memoria (RAM y ROM), varios puertos de entrada y salida (I/O) y periféricos.

Todos estos componentes están interconectados entre sí y en conjunto forman lo que se conoce como microcomputadora.

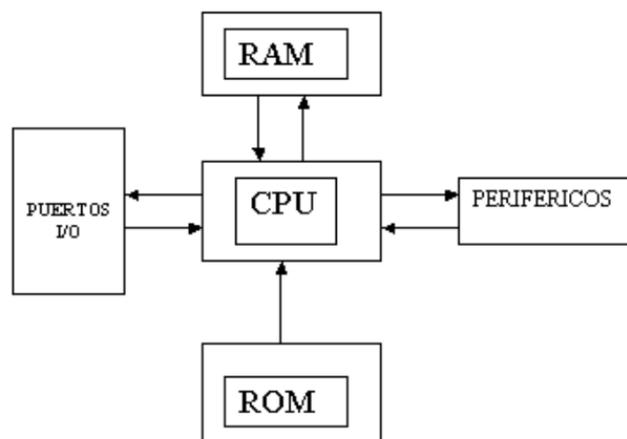


FIGURA 15: ESTRUCTURA INTERNA DEL MICROCONTROLADOR.

Los PIC pertenecen a la familia RISC (*Reduced Instruction Set Computer*) de los controladores con una arquitectura Harvard. Arquitectura mucho más completa que la denominada arquitectura tradicional o arquitectura von Neumann

La arquitectura tradicional se basa en el esquema de John Von Neumann, en la cual la unidad de control de proceso está conectada a una memoria única que contiene todas las instrucciones y datos. El tamaño de la unidad de datos vendrá definido por el ancho

del bus de memoria. En el caso de que un procesador sea de 8-bits y tenga también un bus de 8-bits que lo conecta con la memoria, solamente podrá manejar información de una unidad de longitud (1 byte, 8 bits) en cada acceso a la memoria. Significa esto que siempre que deba acceder a una instrucción o dato complejos en la memoria, mayor que el ancho del bus, deberá llevar a cabo más de un acceso a dicha memoria.

Por otro lado, este bus único limita la velocidad de operación del microprocesador, esto quiere decir que no podrá acceder a la memoria para buscar nueva información antes de que finalice la transferencia resultante del acceso anterior.

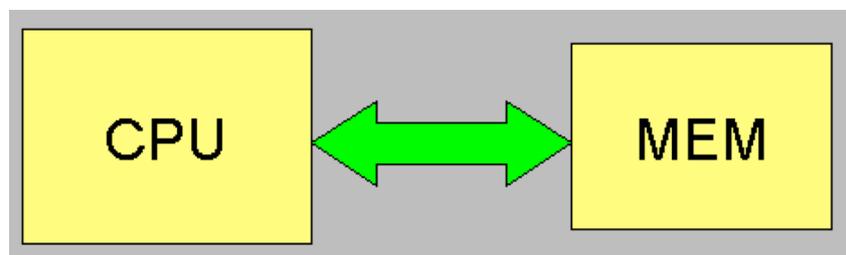


FIGURA 16: ARQUITECTURA TRADICIONAL.

Por otro lado, la arquitectura usada en los PIC's actuales, la Harvard, consiste simplemente en un esquema donde la CPU está conectada por dos buses separados a dos memorias distintas, una que almacena los datos, llamada Memoria de Datos y la otra que almacena las instrucciones, llamada Memoria de Programa.

Por otra parte, al ser buses independientes entre sí, pueden ser de distinto ancho según las exigencias del programa. Para un procesador de Set de Instrucciones Reducido (RISC), el set de instrucciones y el bus que conecta su memoria con la CPU pueden diseñarse para que todas las instrucciones ocupen una sola posición de memoria de programa y por tanto, tener posibilidad de transferir cualquier instrucción con un acceso a la memoria. Además, la independencia entre buses también permite que la CPU

acceda a los datos para completar la instrucción recibida y al mismo tiempo esté leyendo la próxima instrucción a ejecutar.

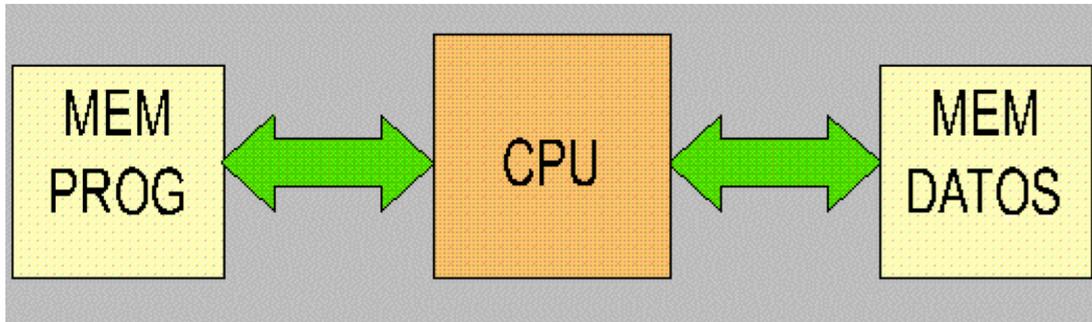


FIGURA 17: ARQUITECTURA HARVARD.

3.4.9. Elección del microcontrolador PIC:

Para el proyecto se ha escogido un PIC de Microchip para controlar y gestionar el BMS debido a las siguientes ventajas:

- Seguridad de acceso: Tienen las memorias aisladas, la de datos y la de programa.
- Protección muy fiable del código.
- Eficiencia del código: Permiten una gran compactación de los programas.
- Rapidez de ejecución: Es capaz de efectuar gran cantidad de operaciones por segundo.
- Herramientas de desarrollo abundantes y con bajo coste.
- Versatilidad: Compatibilidad muy normalizada de pines y código entre los diferentes dispositivos de la misma familia. Además, hay una gran variedad de versiones en distintos dispositivos encapsulados sin una clara reducción de las prestaciones internas.
- Sencillez: Juego reducido de instrucciones y de fácil aprendizaje.

Para facilitar la búsqueda del más apropiado, la página web de Microchip deja al alcance del usuario una amplia gama de controladores ordenados y filtrados por prestaciones. En esta web se pueden seleccionar las prestaciones que ha de tener el PIC y se muestran todos aquellos que las cumplen.

Los requisitos imprescindibles para este PIC son:

- MCU de 16 bits, para poder ejecutar muchas operaciones con números grandes si es necesario.
- 1 puerto USB.
- 1 puerto I²C.
- 1 puerto SPI.
- Cualquier encapsulado que no sea BGA.
- Máxima memoria de programa y datos (SDRAM).
- 32 MHz de frecuencia máxima.
- 24 Canales de ADC, para el control del balanceo de las 16 celdas.
- Alimentación de 3,3V, para que no sea necesario integrar un regulador lineal.

Después de una búsqueda y resultando solamente cuatro modelos con los requisitos mínimos exigidos anteriormente, se escoge el microcontrolador PIC24FJ256DA210 por su amplio número de pines y de I/O, por su rango de alimentación y por sus puertos de comunicación.

La tabla siguiente recopila las propiedades más útiles para este proyecto del microcontrolador escogido:

PROPIEDADES	PIC24FJ256DA210
Arquitectura	Harvard
Procesador	16-bits
Velocidad Procesador	16 MIPS
Memoria de Datos	96KB
Memoria de Programa	256KB
Rango de Temperatura de Operación	-40°C a 85°C
Rango de voltaje de suministro	2,2V a 3,6V
Número de pines	100
Número de entradas y salidas (I/O)	84
Canales ADC	24
Comunicaciones digitales	4 UART (1 USB OTG), 3 I2C y 3 SPI

Tabla 7: Propiedades del Microcontrolador PIC.

Dado que se alimenta a 3,3V, se escoge un convertidor DC/DC aislado de este determinado voltaje, pudiendo con éste alimentar también el sensor hall encargado de medir la corriente de carga y descarga del sistema de baterías.

Componente	Fabricante	Modelo	Rango de Entrada	Potencia
Convertidor DC/DC	Cosel	MGXS6243R3	6V a 60V	5.28W

Tabla 8: Alimentación no aislada.

3.4.10. Sensores de temperatura:

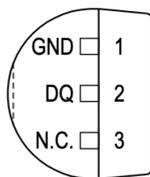
La temperatura es un parámetro muy importante a la hora de gestionar sistemas de baterías. La carga y la descarga de esta misma en una zona indebida de temperatura puede reducir drásticamente la vida de las celdas, Además, las elevadas temperaturas producen un aumento del riesgo de inflamación o incluso explosión. Por tanto, conocer la temperatura en distintas zonas de la batería es un requisito indispensable del BMS en cualquier aplicación con baterías LiFePO₄.

Este BMS puede leer hasta 4 sensores de temperatura ubicados en las zonas donde más probable es el aumento, que son detectados por el controlador de forma automática y periódica.

Una de las configuraciones típicas para este conjunto de sondas es:

- Una sonda para medir la temperatura de los dispositivos de potencia (T1).
- Una sonda para el grupo de celdas del Bloque A (T2).
- Una sonda para el grupo de celdas del Bloque B (T3).
- Una sonda para medir la temperatura ambiente (T4).

SIDE VIEW



TO-92

FRONT VIEW

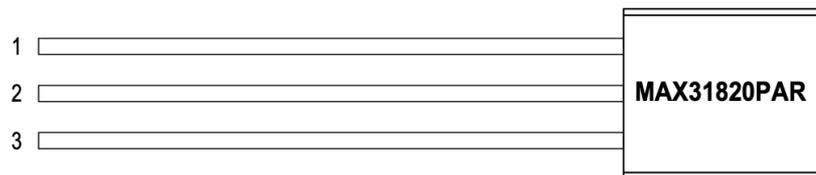


FIGURA 18: SENSOR DE TEMPERATURA.

3.4.11. Elección de los sensores de temperatura:

Con objetivo de simplificar el diseño, se opta por sensores de un solo hilo con alimentación compatible con el controlador. Así, se conecta a uno de los pines de I/O del PIC.

Fabricante	Modelo	Alimentación	Precisión	Resolución	Rango de medida
Maxim Integrated	MAX31820PAR	3V a 3.7V	+/- 0.5°C	9-bit a 12-bit	-55°C a 125°C

Tabla 9: Sensor de temperatura elegido.

Éste se conecta mediante una resistencia pull-up de $4k7\Omega$ conectada a la misma tensión de alimentación que el PIC, y cada uno a un pin distinto.

3.4.12. Medios de comunicación:

Para poder obtener información del dispositivo y poder modificar cualquier parámetro de ajuste, es necesario establecer una comunicación entre un dispositivo controlador y la batería. El medio puede ser propio o puede ser un protocolo ya estandarizado usado en la mayoría de dispositivos de este tipo.

Los protocolos de comunicación más usados son:

RS232: Comunicación serie asíncrona y enlace punto a punto.

Ethernet LAN: Estándar para redes informáticas.

Bus CAN: Estándar para redes en automoción.

USB: Estándar para periféricos de ordenadores y enlaces punto a punto.

De todos los anteriormente explicados, para el proyecto se implementa la conexión USB (Universal Serial Bus), ya que este tipo de comunicación permite conectar el dispositivo a cualquier ordenador.

Además de ésta, también se implementa una posibilidad de conexión I2C, la que se empleará para posibles conexiones de comunicación internas del dispositivo.

Protocolo USB:

Como se puede ver en la hoja de datos de microcontrolador PIC, éste tiene la posibilidad de configurar un USB 2.0 On-The-Go, sin embargo este tipo de puertos necesita un firmware y un driver. Por ello, se opta por usar un conversor USB-Serie, ya que en este tipo de convertidores, tanto el driver como el firmware ya vienen integrados.

Este dispositivo se conectará directamente a una de los pines UART (Universal Asynchronous Receiver-Transmitter) del PIC. Esto provocará que al encender el pc de la plataforma móvil en la que se integre este dispositivo, éste lo reconozca como un puerto serie estándar y quede añadido al sistema. Así, el ordenador puede acceder a él como si fuera un puerto serie convencional.

El convertidor elegido para esta aplicación es el siguiente:

Fabricante	Modelo	Alimentación	Nº Pines	Nº GPIO	Corriente de Salida
Silicon Labs	CP2102N-A02-GQFN20	3V - 3,7V	20	6	100mA

Tabla 10: Convertidor USB elegido.

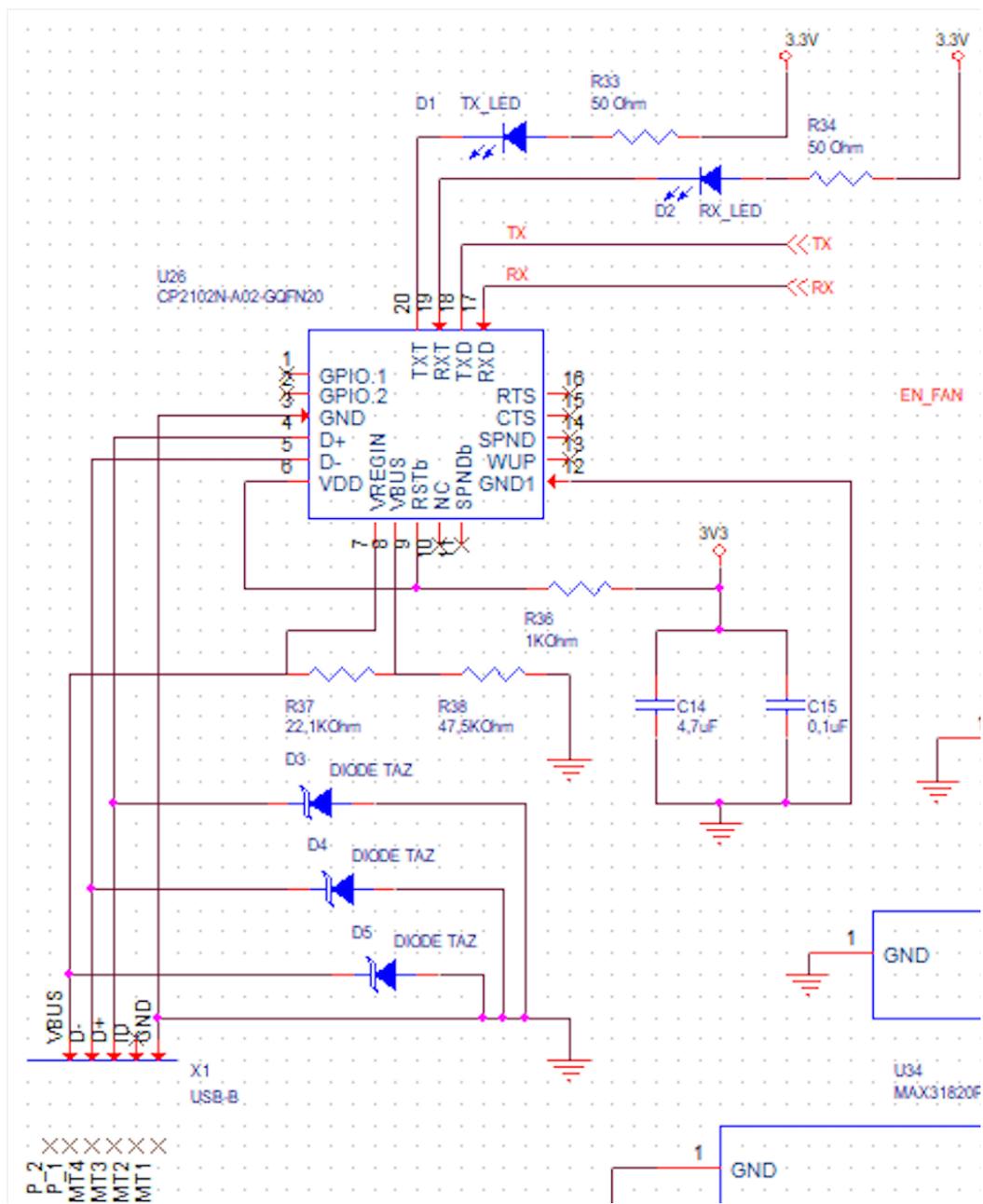


FIGURA 19: CONEXIONES USB.

Interfaz I2C:

Éste es un puerto y protocolo de comunicación serie, define la trama de datos y las conexiones físicas para transferir bits entre dos dispositivos digitales. El puerto incluye dos cables de comunicación SDA (Serial Data) y SCL (Serial Clock).

Permite la conexión de hasta 127 dispositivos esclavos en esas dos líneas, con hasta velocidades de 100, 400 y 1000 kbits/seg.

Éste es uno de los más utilizados para comunicarse con sensores digitales, ya que a diferencia del puerto serie, su arquitectura permite tener una confirmación de los datos recibidos dentro de la misma trama.

Para este protocolo se diferencian dos elementos básicos, el “maestro” y el “esclavo”, y la conexión típica entre diferentes dispositivos con este protocolo sería la siguiente:

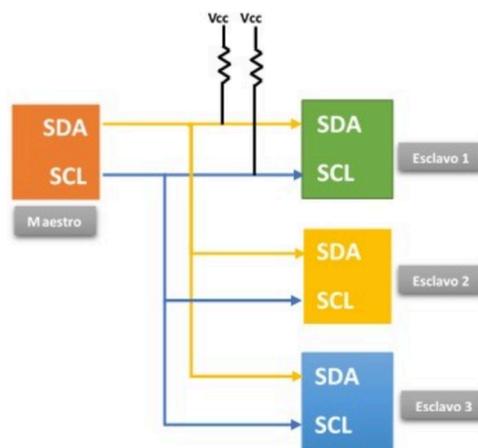


FIGURA 20: CONEXIÓN ENTRE DISPOSITIVOS MAESTRO-ESCLAVO.

Dado que el PIC designado para el proyecto (PIC24FJ256DA210) tiene 3 puertos I2C que pueden funcionar como maestro o como esclavo, el BMS se configura como esclavo y el Pc de la propia plataforma móvil se designará como maestro. Para poder realizar la conexión apropiada, es suficiente con seguir el esquema anterior.

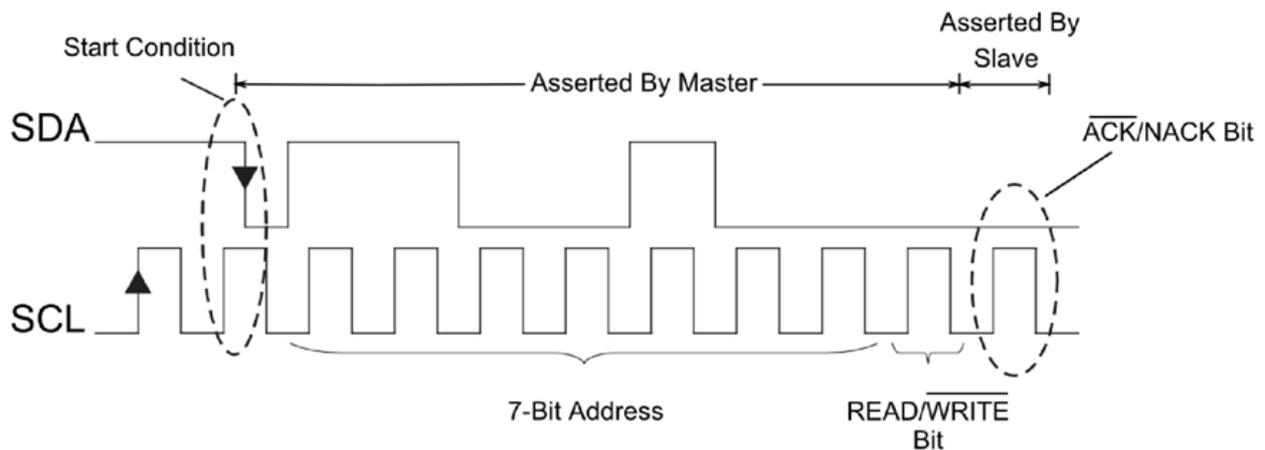


FIGURA 21: TRANSMISIÓN DE UN BYTE.

El orden de las transferencias para una comunicación estable y efectiva es el siguiente:

- Es necesaria una dirección para la identificación de cada dispositivo. La dirección estándar de I2C se define en el primer byte enviado por el maestro, para especificar si el esclavo debe recibir datos del maestro o enviarlos al maestro.
- Por la línea SDA se envía el bit "Start" por el maestro con la dirección del esclavo acompañada del bit de R/W a 0, indicando la función escritura.
- El esclavo envía el bit de ACK (Acknowledge) como reconocimiento.
- La dirección del registro deseado es enviada por el maestro y el esclavo la confirma con el bit ACK otra vez.
- El maestro envía por segunda vez el bit "Start" seguido de la dirección del esclavo con R/W a 1, para indicar lectura.
- El esclavo reconoce la lectura y transmite los datos al maestro.
- El maestro envía una señal de ACK al esclavo cada vez que termina de leer un byte para indicar al esclavo que envíe más bytes.
- Una vez todos los datos enviados por el esclavo y leídos por el maestro, el maestro envía una señal de NACK (negative acknowledgement) y recupera el control de la línea SDA con el bit de parada "Stop".

3.4.13. Resto de conexiones:

Los indicadores luminosos necesarios para obtener la información adecuada visualmente desde la placa de gestión de baterías irán conectados al PIC a través de una resistencia limitadora.

La barra indicadora del estado de la batería consta de nueve led's divididos en tres grupos. El primer grupo, con iluminación roja, indica carga muy baja, menor del 20% de la capacidad total de la batería; el segundo grupo, con iluminación amarilla, indica carga media, menor a 80% de carga total; y el tercer grupo, con iluminación verde, indica carga máxima.

Cuando la batería esté entre el límite mínimo de capacidad y el 20%, el grupo de led's rojos indicará este estado con un parpadeo simple. Una vez la carga sobrepase el 20%, el grupo de led's rojos pasará a un estado de iluminación fija hasta que termine el ciclo de carga, siendo el segundo grupo, de iluminación amarilla el que entre en el estado de parpadeo simple. Del mismo modo, cuando la capacidad de la batería sobrepase el 80%, el grupo de led's amarillos pasará a un estado fijo y el grupo de led's verdes pasará al estado de parpadeo simple, solamente cuando se realiza todo el ciclo de carga y la capacidad de la batería es de un 100%, los tres grupos de led's, el rojo, el amarillo y el verde, iluminarán en un estado fijo continuo hasta que se deshabilite la carga.

El control de cada grupo se llevará a cabo con un MOSFET (BS170), dado que las salidas de corriente y tensión suministrados por el PIC son insuficientes para iluminarlos.

Además, el ventilador de la etapa de potencia se controla a través de otro MOSFET, el mismo modelo que el anterior, quedando completamente accesible un conector hembra para el ventilador.

También, el encendido de la batería se realiza mediante un interruptor, cuyo cableado conecta a masa en caso de apagado y a 3,3V en caso de encendido.

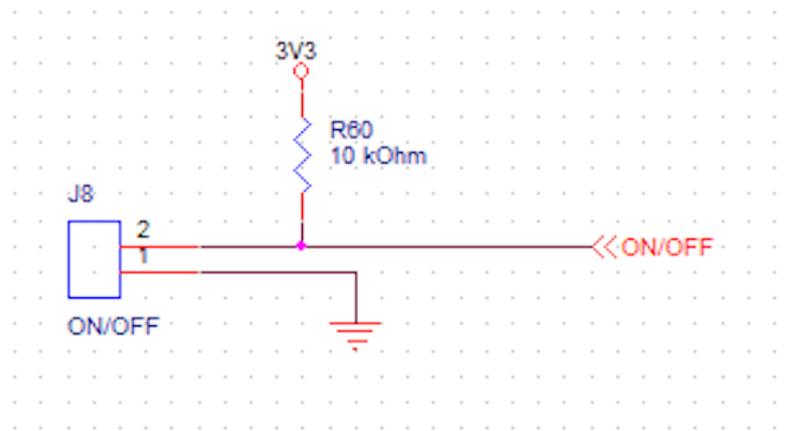


FIGURA 22: CONEXIÓN DEL INTERRUPTOR ON/OFF.

3.5. TARJETA DE POTENCIA:

El dispositivo de gestión de baterías tiene que controlar tanto la carga como la descarga y garantizar la apertura cuando la agrupación de celdas se sale del área de operación segura (SOA).

Este aspecto se controla mediante dos MOSFETs, uno para la línea de carga y otro para la línea de descarga. Estos dos funcionan como un interruptor bidireccional, y para ello, tienen que trabajar en la zona de corte y en la zona de saturación.

El funcionamiento de este pequeño circuito es el siguiente:

Si los dos MOSFETs están en la zona de corte, significa que la tensión no puede circular en ninguno de los dos sentidos, y con los dos en estado de saturación se permite en ambos.

Con el MOSFET de carga en saturación y el otro en corte, se permite exclusivamente la circulación de la corriente por la línea de la carga en dirección a la batería, ya que el cargador está a mayor potencial que la batería y el diodo del MOSFET de descarga permite la circulación hacia la batería.

Con el MOSFET de descarga en saturación y el otro en corte, situación inversa a la explicada anteriormente, solo se permite la descarga, ya que la batería está a mayor potencial que la carga y el diodo del MOSFET de carga permite la circulación de corriente hacia el exterior.

Con esto, cuando el interruptor de la batería esté en “OFF” o se produzca un fallo, el controlador desactivará los dos MOSFET's. Por el contrario, si no hay ningún fallo y el

interruptor está en “ON”, el controlador activará los dos MOSFET’s permitiendo tanto la carga como la descarga.

MODO	CHG MOSFET	DCHG MOSFET	CORRIENTE
Apagado	OFF	OFF	Ninguna
Espera	ON	ON	Ninguna
Carga	ON	OFF	Positiva
Descarga	OFF	ON	Negativa

Tabla 11: Estados de los MOSFET.

3.5.1. Cálculo del MOSFET de potencia:

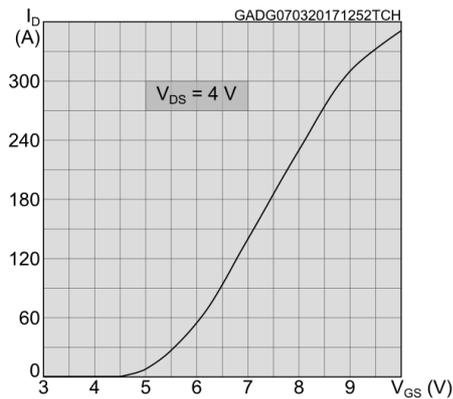
Se busca un MOSFET cuya resistencia en conducción (R_{on}) muy baja, y su tensión máxima sea superior a 60V (V_{ds}), una corriente de drenaje (I_d) superior a los 50A, una tensión umbral de puerta baja (V_{gs}) para poder controlarse con las salidas digitales del controlador y un encapsulado TO-220.

Cumpliendo estas características existen, entre otros, estos MOSFET’s.

Modelo	V_{ds}	I_d	$V_{gs\ th}$	V_{gs}	R_{on}
STP130N8F7	80 V	80 A	2,5 V	20 V	5,8 m Ω
FDP036N10A	100 V	75 A	2 V	20 V	3,6 m Ω
SUP90N06-6M0P	60 V	90 A	2,5 V	20 V	6 m Ω
IPP020N06N	60 V	120 A	2,1 V	20 V	2 m Ω

Tabla 12: MOSFET’s adecuados.

De los cuatro definidos en la tabla, se escoge el primero, STP130N8F7, por presentar una versatilidad y un buen compromiso entre todos sus parámetros.



En esta curva se puede observar que cuanto mayor es V_{GS} , mayor rendimiento presenta el transistor MOSFET, menor R_{on} y mayor I_D .

FIGURA 23: COMPORTAMIENTO DEL MOSFET STP130N8F7.

Por tanto, la puerta de los MOSFET's se activará con un pulso de 15V alimentado desde una fuente de alimentación conmutada aislada RJZ-3.315S/HP.

El diodo parásito entre drenador y fuente (source-drain diode) es capaz de soportar la misma corriente que el transistor (80A), pero todos estos suelen tener una recuperación bastante lenta. Esto provoca picos de corriente de recuperación inversa que pueden destruir el dispositivo, por ello, se recomienda sustituirlo por un diodo externo rápido. En este caso, se añade en paralelo con cada MOSFET un diodo de conmutación ultrarrápido MUR460.



FIGURA 24: LED PARÁSITO ULTRARRÁPIDO.

Por último, los dos MOSFET's se adhieren a un disipador con ventilación incorporada, modelo de Fischer Elektronik, LAM 3 K 100 05. Dicho ventilador será accionado por una de las salidas del PIC.

Por otra parte, las salidas del PIC que activan los MOSFET's serán aisladas por un aislador digital modelo ADM3210TRZ-EP, dado que tiene un umbral de voltaje de entrada, compatible con el PIC de entre 3,3V y 5V y un margen de voltaje de salida de entre 4,5V y 18V, compatible con la fuente conmutada aislada que activará los MOSFET's, ISH0515A.

3.5.2. Sensor de corriente efecto HALL:

El sensor Hall, como su propio nombre indica, es un dispositivo cuyo funcionamiento se basa en el efecto Hall para llevar a cabo detecciones y mediciones de campos magnéticos o corrientes.

Al fluir corriente por un sensor de este tipo y aproximarse un campo magnético que fluye en dirección al eje vertical del sensor, éste crea un voltaje saliente proporcional al producto de la fuerza del campo magnético por la corriente. Si se conoce el valor de la corriente, se puede calcular la fuerza del campo magnético, si se crea un campo magnético concreto por medio de una bobina o un conductor, se puede medir el valor de la corriente en dicha bobina o conductor.

Sus propiedades dimensionales suelen ser pequeñas, ya que no presenta una estructura compleja. Tiene una exactitud de medición muy alta, junto con una alta linealidad de la sensibilidad y estabilidad de la temperatura. En este proyecto se usa una sonda Hall para medir la corriente de carga o de descarga de la batería instantáneamente.

Las sondas Hall encontradas y que podrían adaptarse al dispositivo, son las siguientes:

Modelo	Corriente	Sensibilidad	Temperatura	Sobrecorriente	Error
ACS781LLRTR-050B-T	-50 a +50 A	26.4 mV/A	-40 a +150 °C	600 a 1200 A	0.8%
ACS781LLRTR-100B-T	-100 a +100 A	13.2 mV/A	-40 a +150 °C	600 a 1200 A	0.8%
ACS781LLRTR-150B-T	-100 a +100 A	8.8 mV/A	-40 a +150 °C	600 a 1200 A	0.8%

Tabla 13: Sensores Hall adecuados.

De las tres sondas Hall de la tabla anterior, se escoge ACS781LLRTR-100B-T, dado que las plataformas móviles a las que se va a intentar integrar este dispositivo pueden tener picos de consumo de más de 50 A, pero es muy improbable que lo tengan de más de 100 A. En cualquier caso, estas sondas son intercambiables, pudiendo sustituirse una por otra sin problemas si es necesario.



FIGURA 25: SENSOR HALL.

4. DISEÑO ELECTRÓNICO EN OrCAD:

4.1. SOFTWARE OrCAD:

OrCAD es un software propio, en su versión 17.4, utilizado para la Automatización de Diseño Electrónico (EDA). El software es usado por ingenieros para diseñar esquemas electrónicos, simulación electrónica y esquemas de circuito impreso para poder fabricar dichas tarjetas (PCB).



FIGURA 26: LOGO SOFTWARE ORCAD.

OrCAD fue asumida en 1999 por Cadence Design Systems y se integró con Cadence Allegro en 2005.

OrCAD es un conjunto de producto para diseñar y analizar las tarjetas de circuitos impreso, el cual incluye un editor de esquemas como el Capture CIS, un convertidor analógico y de señal mixta simulador de circuitos como el PSpice y una solución de diseño de la placa de circuito impreso como PCB Editor.

OrCAD ha sido el software elegido para el diseño del BMS por ser uno de los más versátiles en el aspecto de recopilación de información y de uso de archivos creados en los módulos anteriores en sus módulos correspondientes más avanzados. Además, un

usuario cualquiera tiene acceso a una versión gratuita de prueba en la que siguen operando muchos módulos y se mantienen muchas de las soluciones que facilita el software completo.

4.1.1. OrCAD Capture CIS:

OrCAD Capture CIS puede crear esquemas electrónicos para diseños analógicos o mixtos, diseños de placas de circuitos impresos y diseños de lógica programable. El editor de páginas de esquemas de Capture ayuda a crear el diseño de la forma más eficaz.

Capture se comunica de forma interactiva con OrCAD PSpice o OrCAD Layout, de modo que puede depurar fácilmente los proyectos.

Además de las 30.000 librerías incluidas, puede crear sus propios componentes y librerías.

4.1.2. OrCAD PSpice:

OrCAD PSpice A/D simula circuitos analógicos, digitales y mixtos. Los algoritmos tanto analógicos como digitales de PSpice A/D están incluidos dentro del mismo programa de modo que pueden simularse circuitos mixtos analógico/digitales con bucles de realimentación fuertemente conectados entre secciones digitales y analógicas sin ninguna reducción en las prestaciones.

Después de preparar un diseño para su simulación, OrCAD Capture genera un conjunto de ficheros del circuito. Este conjunto de ficheros, conteniendo la lista de conexiones del circuito y los comandos de análisis, es leído por PSpice A/D para su simulación. PSpice A/D los formula en gráficas significativas, que pueden ser marcadas para representarse directamente desde las páginas del esquema por medio de marcadores.

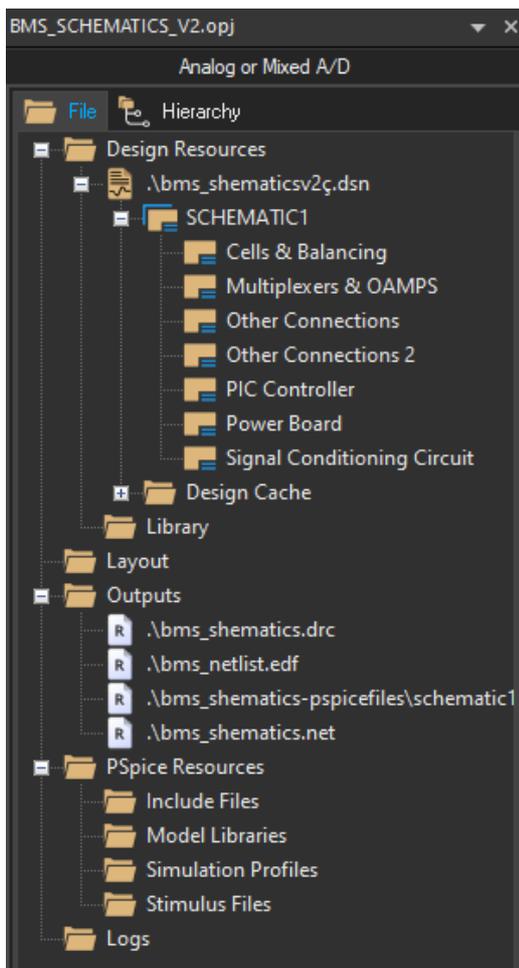
4.1.3. OrCAD PCB Editor (Layout):

Permite el diseño de PCB ayudándose de los archivos de los productos anteriores (Capture y PSpice) o ficheros de otros programas, su objetivo es facilitar al usuario la creación del circuito con todos sus componentes, su ubicación o emplazamiento sobre la placa, la interconexión de los dispositivos electrónicos, la generación de máscaras, la recopilación de la documentación y la exportación de los datos.

4.2. DISEÑO DE ESQUEMAS ELECTRÓNICOS:

Una vez con todos los componentes definidos y descritos en el apartado 3 de este proyecto “Diseño Hardware”, se implementan los esquemas electrónicos de cada una de las partes e interconexiones de la placa de circuito impreso.

Una vez en la propia interfaz y teniendo creado el proyecto nuevo de nombre “BMS_Schematics”, se estructura el gestor de proyectos presente a la izquierda del Workspace, cuyo objetivo es dar una representación gráfica de los recursos de un proyecto agrupándolos en las carpetas apropiadas descritas a continuación:



- La carpeta Design Resources es la que agrupa, como su propio nombre indica, todos los recursos de diseño necesarios del proyecto.
- A continuación, la carpeta esquemas, donde quedan ordenados del primero al último.
- La carpeta Design Cache, en la que se muestran todas las partes utilizadas en las páginas de esquemas, cualquier carpeta de esquema o página de esquema será añadida a esta carpeta.
- La carpeta Library, muestra los ficheros de librerías de partes del esquema que se han añadido al proyecto utilizando el comando Project en el menú Edit.

FIGURA 27: VISOR DE PROYECTOS DE ORCAD.

- La carpeta Outputs, muestra la salida de las herramientas de proceso en Capture. Generalmente estos ficheros incluyen listas de materiales y listas de conexiones específicas de cada tecnología. Capture añadirá los ficheros apropiados a esta carpeta cada vez que se generen.

Cada proyecto puede tener sólo un diseño, pero puede tener múltiples librerías. El diseño puede consistir en cualquier número de esquemas o modelos VHDL, pero deberá tener un único módulo principal. El módulo principal está definido como el nivel principal de diseño, esto quiere decir que el resto de módulos estarán referenciados dentro del módulo principal.

En el caso del proyecto presente, se cuenta un total de siete páginas de esquemas, las cuales presentan todas las conexiones existentes en las dos placas de circuito impreso, la de control y la de potencia. Están presentes las conexiones de las diferentes alimentaciones, los circuitos de medida, los circuitos de balanceo, los circuitos de acondicionamiento de señal, los circuitos de aislamiento, el circuito de comunicación USB OTG, el circuito de potencia, el circuito de marcha/paro y el circuito para las luces LED.

Esquema 1 - "Cells & Balancing":

Incluye dos conectores de ocho pines donde se conecta el voltaje de control de cada una de las celdas para su correspondiente medición.

Incluye el circuito de balanceo de cada celda, compuesto por una señal de salida del PIC para activar cada MOSFET. Una resistencia en serie al propio MOSFET para efectuar el balanceo y una conexión al negativo del microcontrolador para sensibilizar la señal.

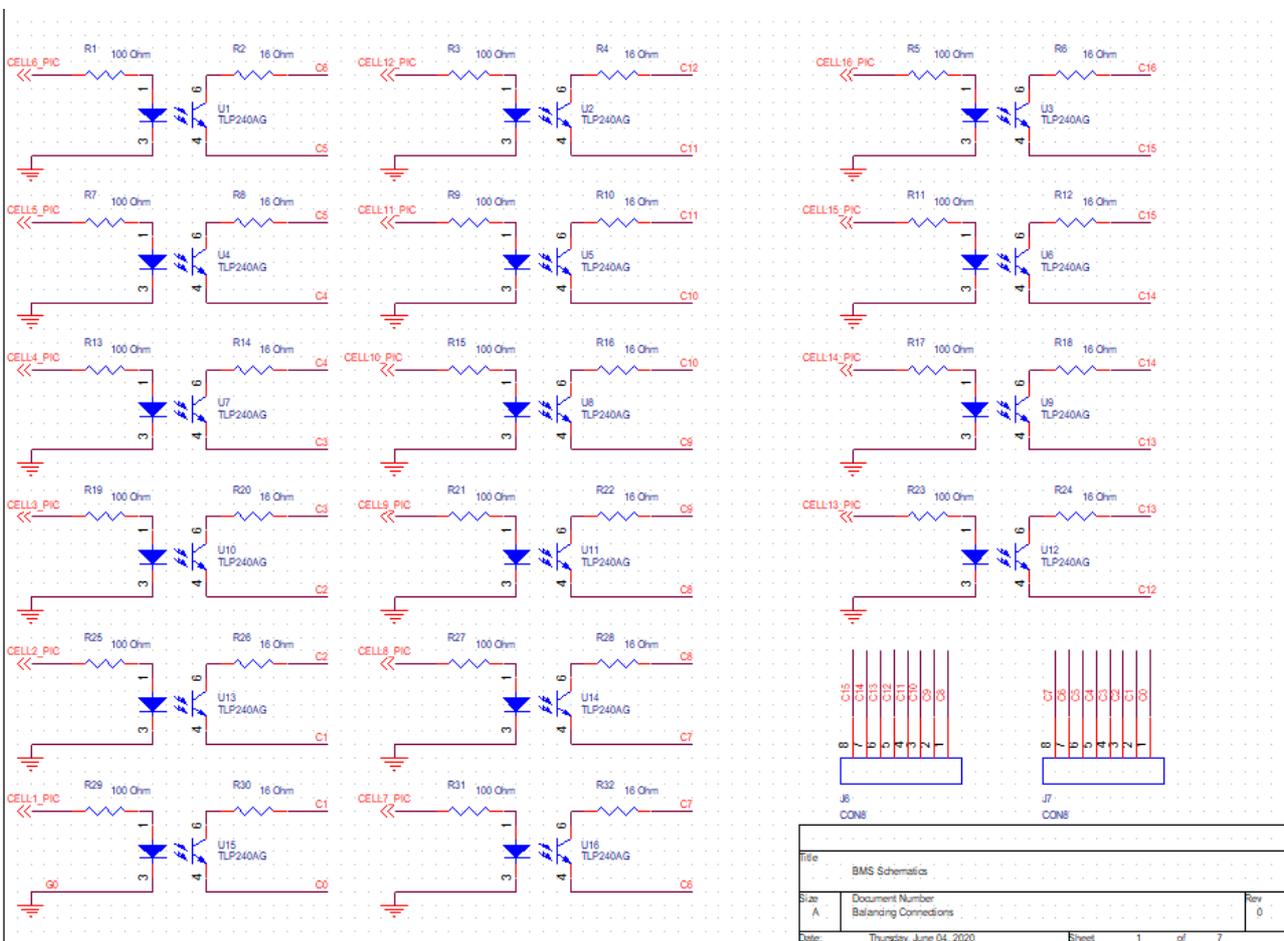


FIGURA 28: ESQUEMA DE BALANCEO

Esquema 2 - "Multiplexers & OAMPS":

Incluye las conexiones de los voltajes de control de cada uno de los bloques de celdas a los multiplexores encargados de comparar cada uno de ellos y enviar una señal de salida adecuada hacia el controlador del balanceo.

Incluye las conexiones de las señales de entrada y de salida de los aisladores digitales necesarios para poder sensibilizar la señal de "elección de celda" emitida por el PIC al voltaje de los multiplexores.

Incluye un amplificador diferencial en cada uno de los dos circuitos para facilitar su acondicionamiento en esquemas siguientes.

Incluye un convertidor dc/dc de 3,3V aislado de la tensión de los multiplexores y de las celdas de batería para poder alimentar, además de la parte del controlador PIC y el sensor de corriente, una de las entradas de alimentación de los aisladores digitales.

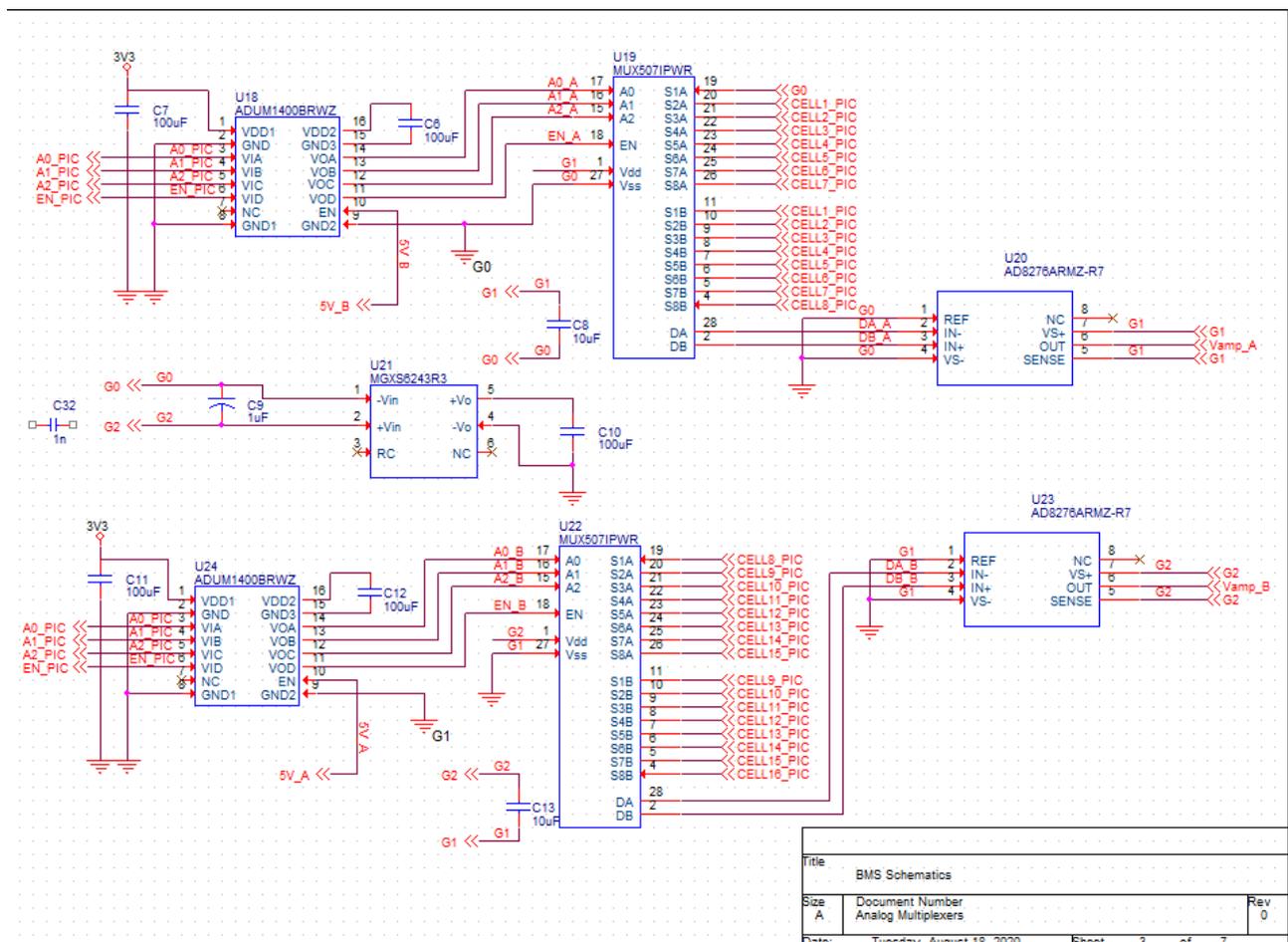


FIGURA 29: CONEXIONES MULTIPLEXORES.

Esquema 3 - "Other Connections":

Incluye el circuito necesario para que el controlador USB2.0 CP2102N-A02-GQFN20 convierta la señal de UART a USB, debido a que el microcontrolador sólo tiene la posibilidad de conectar un USB2.0 On-The-Go y éste necesita un firmware y un driver para su control. Por tanto, se conectarán sus señales a dos señales UART del PIC.

Incluye la alimentación y la señal de salida de los cuatro sensores de temperatura.

Señales de referencia que se transmitirán hasta el microcontrolador para su posterior diagnóstico.

Incluye el conector y el MOSFET que activaría el ventilador refrigerador en caso de que fuera necesario.

Incluye las conexiones y los MOSFET usados para controlar la barra de led's indicativos de la carga de la batería.

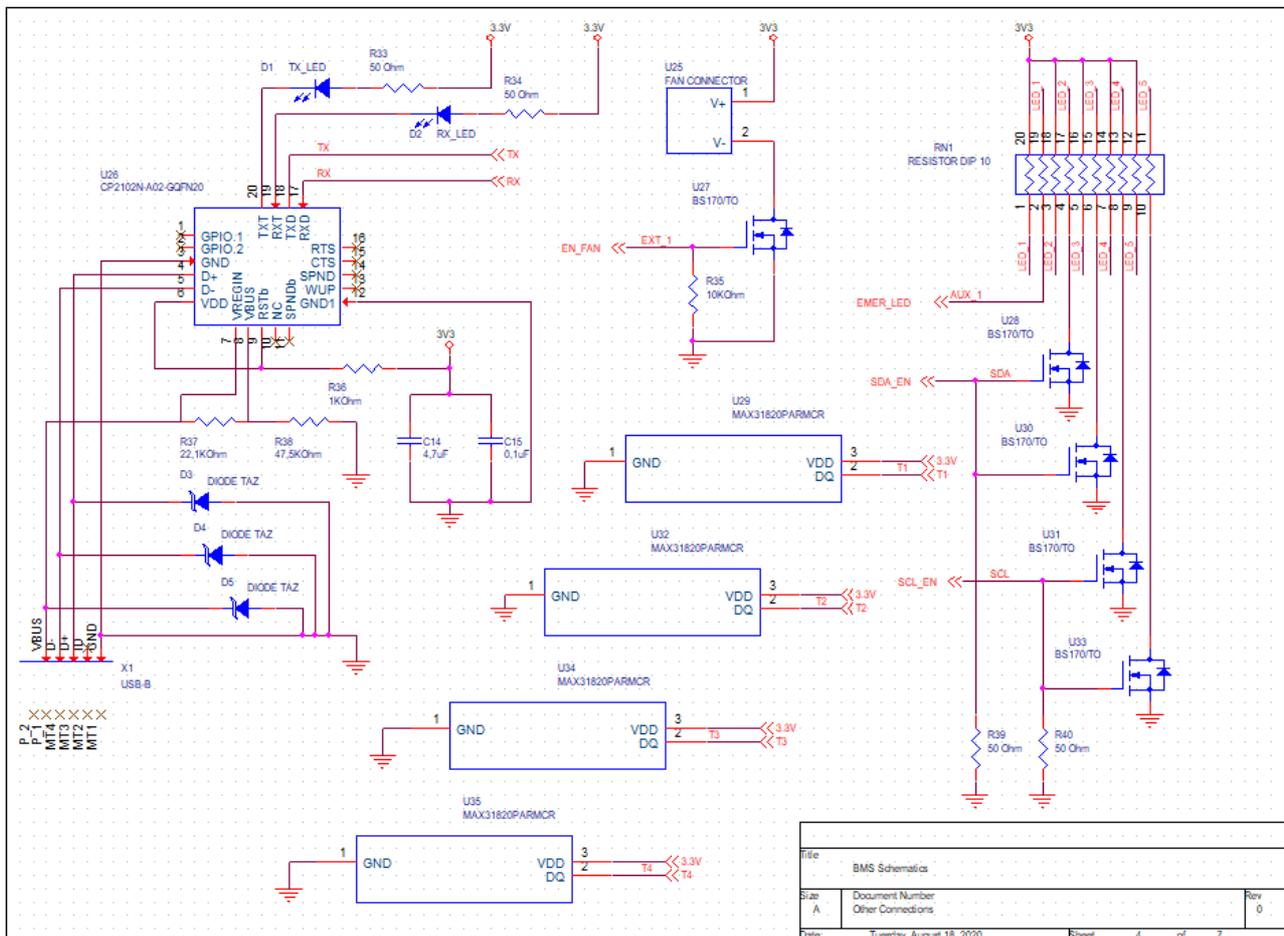


FIGURA 30: CONEXIONES USB, SENSORES Y LED

Esquema 4 - "Other Connections 2":

Incluye el conector de alimentación y comunicación que va a ser necesario implementar para que la tarjeta de potencia pueda estar referenciada a los voltajes principales de la tarjeta de control.

Incluye un amplificador diferencial para adaptar al rango de la entrada analógica a la señal del sensor de corriente efecto Hall.

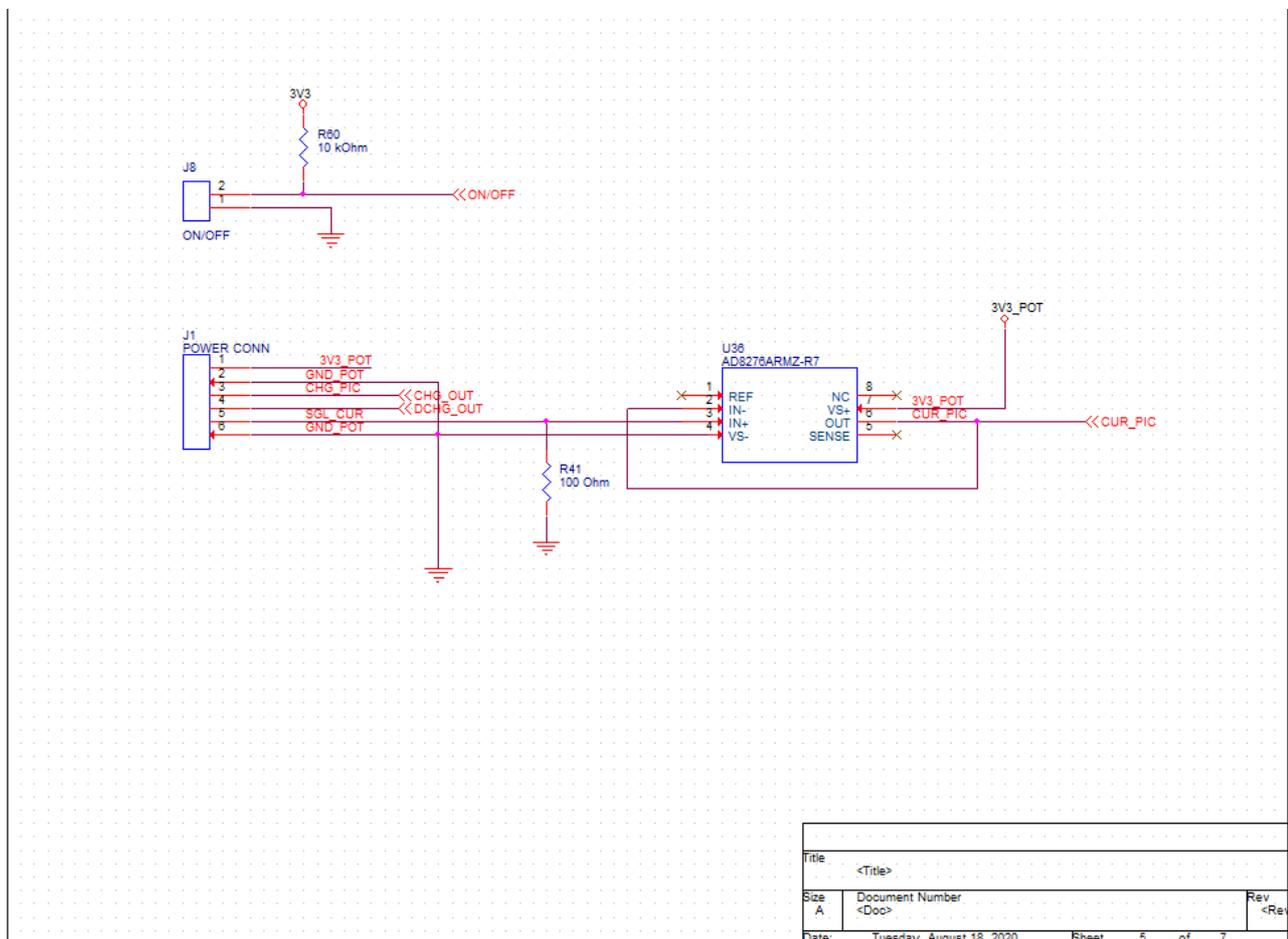


FIGURA 31: CONEXIONES ENTRE PLACAS Y ON/OFF

Esquema 5 - "PIC Controller":

Incluye el microcontrolador PIC24FJ256DA210 del fabricante Microchip con las conexiones de alimentación y comunicación necesarias para llevar a cabo la gestión del dispositivo conectado.

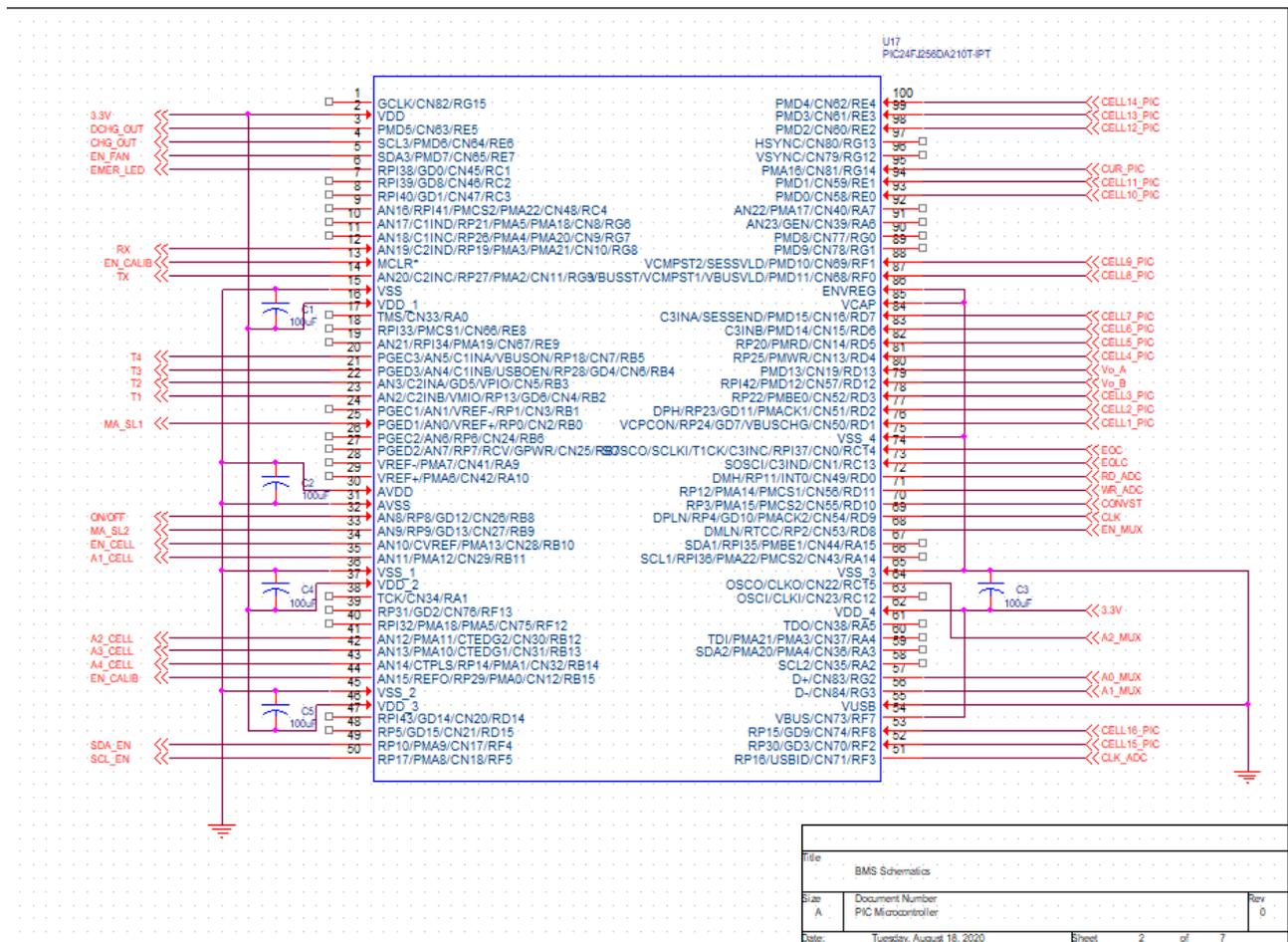


FIGURA 32: CONEXIONES DEL MICROCONTROLADOR PIC

Esquema 6 - "Power Board":

Incluye todo lo relativo a la tarjeta de potencia del BMS.

Incluye el conector originado en la tarjeta de control en el que se incluye la alimentación de los dispositivos de la tarjeta de potencia y el retorno de la señal de medición de corriente del sensor efecto Hall.

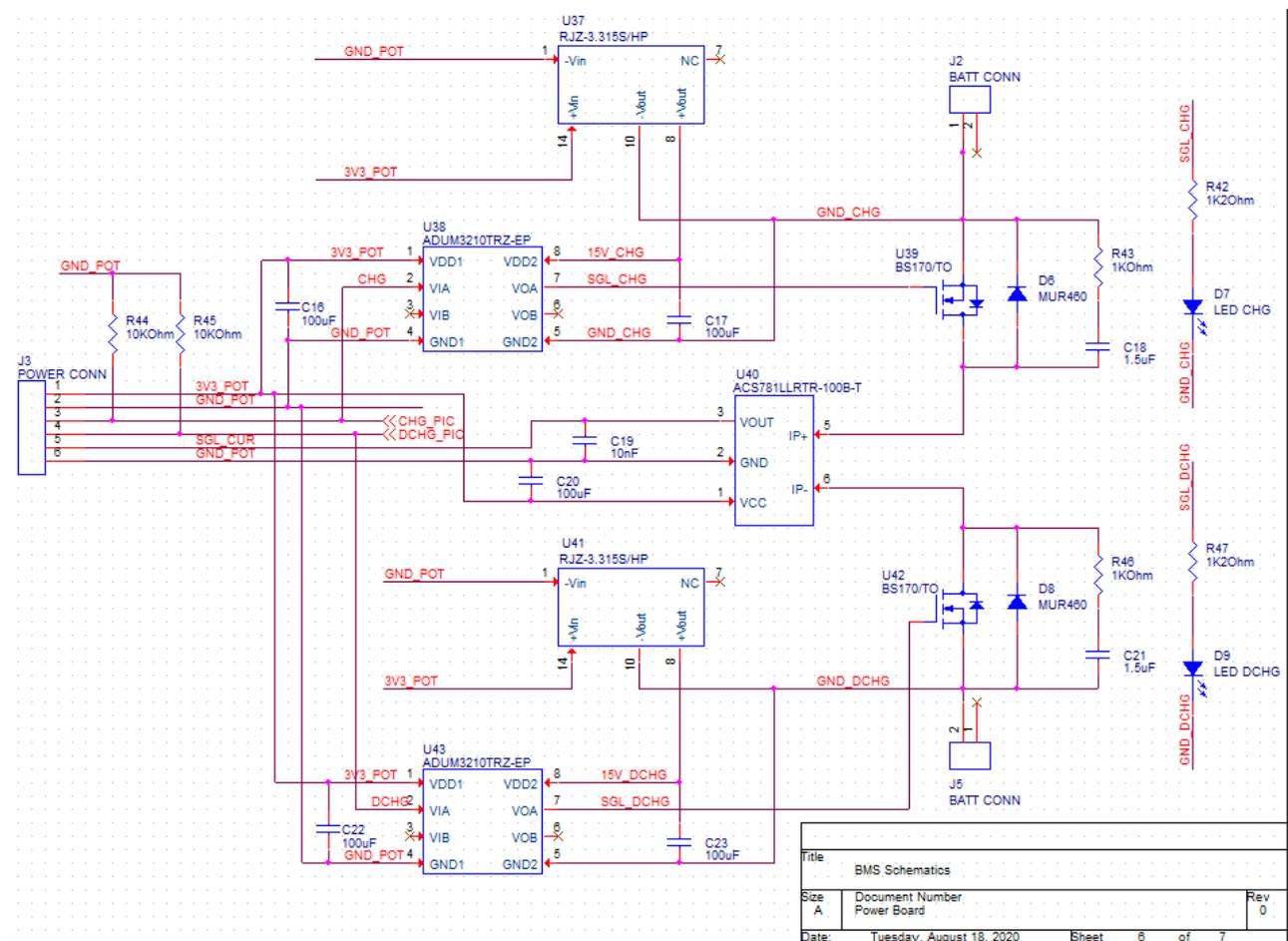


FIGURA 33: CONEXIONES DE LA PLACA DE POTENCIA

Esquema 7 - "Signal Conditioning Circuits":

Incluye los dos circuitos de acondicionamiento de señal necesarios para obtener un rango de voltaje compatible con el ADC del microcontrolador.

Los componentes incluidos en cada uno de los circuitos de acondicionamiento son: la de tensión de 3V, el seguidor de tensión a la salida de la referencia de voltaje, el amplificador diferencial o restador, el amplificador no atenuador con ganancia cuatro y el amplificador de aislamiento.

Incluye también un convertidor dc/dc no aislado para las alimentaciones de todos los amplificadores operacionales, excepto la parte aislada del amplificador operacional de aislamiento.

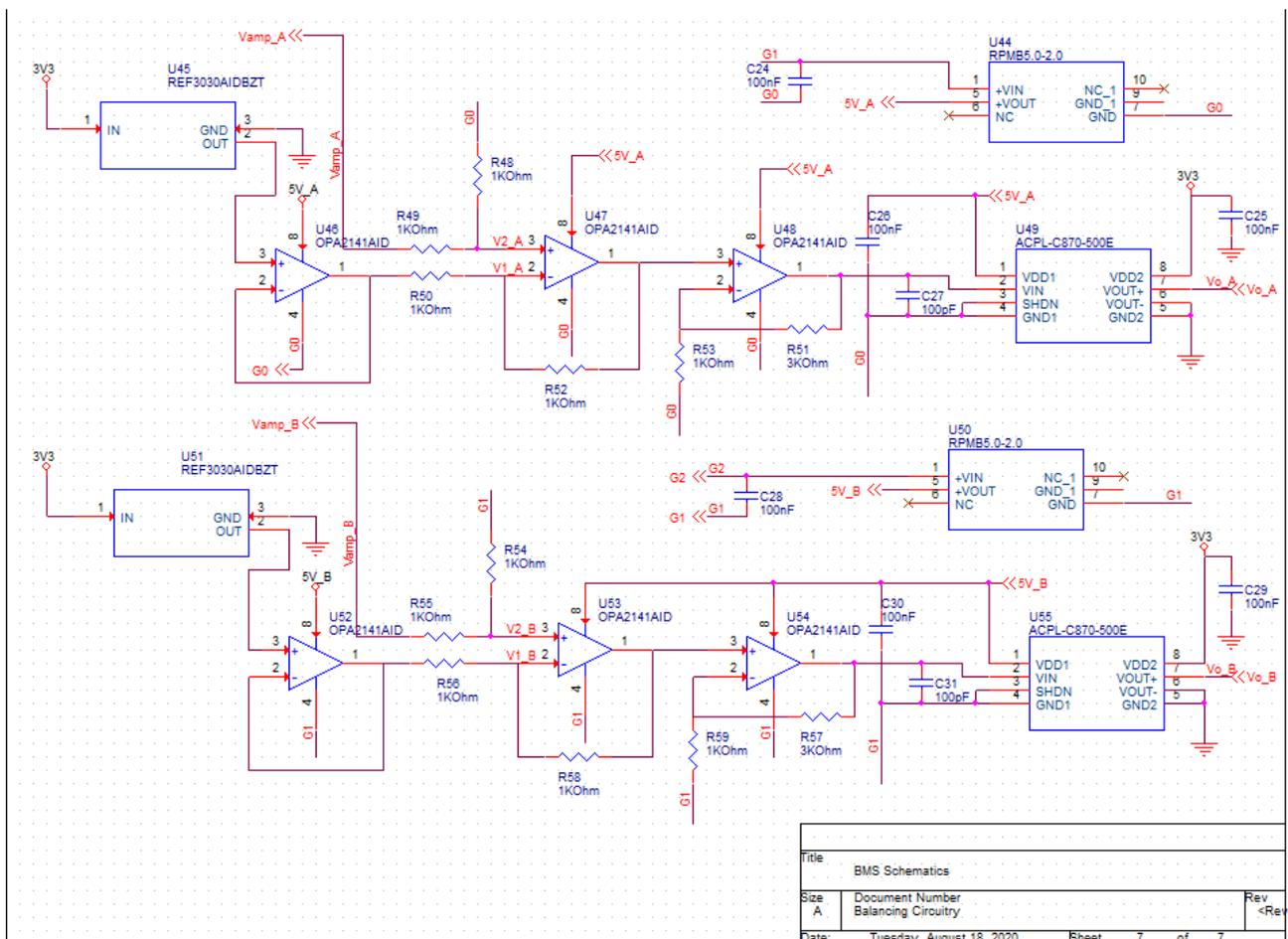


FIGURA 34: CIRCUITO DE ACONDICIONAMIENTO DE SEÑAL

5. DISEÑO SOFTWARE:

El dispositivo diseñado en el proyecto incluye un microcontrolador que es preciso programar para que, mediante una máquina de estados, adopte el comportamiento adecuado delante de todas las situaciones que se pueden registrar en base a un sistema de baterías LiFePO₄. El PIC es el encargado de gestionar el comportamiento del BMS a partir de una lectura periódica de todos los sensores integrados tanto en la placa de control, como en la de potencia, como en los sensores repartidos por la distribución de celdas del sistema de baterías, concretamente, los sensores de temperatura.

5.1. PROGRAMACIÓN DEL PIC:

La rutina principal consiste en un bucle infinito en el que se implementa una máquina de estados como la que se contempla en la página siguiente. Esta técnica permite que sea más fácil no cometer errores en la programación y que el programa no se quede “atrapado” en cualquiera de las funciones auxiliares.

Por ser un programa largo y complejo, se elige programar estructuralmente.

Dentro del proyecto llamado LiFePO₄_BMS.X, se han creado distintos ficheros .c y .h para compartir y aunar todas las librerías, definiciones, funciones y variables necesarias para que el programa pueda llevar a cabo una medición periódica del estado de la batería y pueda ejecutar un diagnóstico en un corto espacio de tiempo para reaccionar ante dichos cambios de estado.

Los ficheros incluidos en este proyecto son 16 en total. Ocho ficheros .c y ocho .h:

5.1.1. Main.c:

Contiene la rutina principal, es decir, el programa madre donde se llama a las funciones auxiliares necesarias, tanto del mismo fichero como de cualquiera del proyecto. Este bucle, como ya se ha explicado anteriormente, está siempre en continuo funcionamiento, concretando así los modos de funcionamiento del BMS para cada ocasión (carga, descarga y espera), funciones genéricas tanto de habilitación como de deshabilitación y funciones de verificación de límites y datos.

5.1.2. Initialize.c:

Contiene todas las variables, constantes y señales que se lanzan con la inicialización del BMS. Contiene también las funciones que determinan la configuración concreta del oscilador y de los puertos.

5.1.3. Initialize.h:

Contiene las constantes que se declaran globales para todo el programa. Además, también contiene las declaraciones de las funciones que se utilizan para la lectura y establecimiento de datos del fichero Initialize.c para globalizar las variables entre todos los ficheros.

5.1.4. Balancing.c:

Contiene las funciones correspondientes al control de balanceo de las celdas.

5.1.5. Balancing.h:

Contiene las constantes que se declaran globales para todo el programa. Además, también contiene las declaraciones de las funciones que se utilizan para la lectura y establecimiento de datos del fichero Balancing.c para globalizar las variables entre todos los ficheros.

5.1.6. Auxiliar.h:

Contiene variables globales y constantes relacionadas con el programa, menos el apartado I2C. También contiene las declaraciones de las funciones del fichero main.c.

5.1.7. Readsymux.c:

Este fichero contiene la función de selección de los multiplexores y las funciones relacionadas con la lectura de datos.

5.1.8. Readsymux.h:

Contiene las constantes que se declaran globales para todo el programa. Además, también contiene las declaraciones de las funciones que se utilizan para la lectura y establecimiento de datos del fichero Readsymux.c para globalizar las variables entre todos los ficheros.

5.1.9. Communications.c:

Éste contiene todas las funciones relacionadas con las comunicaciones, incluyendo la inicialización del I2C, la función de comunicación USB, la configuración de los puertos UART y una función para calcular el tiempo de respuesta según la función que se esté ejecutando.

5.1.10. Thermos.c:

Contiene todas las funciones relacionadas con los sensores de temperatura.

5.1.11. Thermos.h:

Contiene las constantes que se declaran globales para todo el programa. Además, también contiene las declaraciones de las funciones que se utilizan para la lectura y establecimiento de datos del fichero Thermos,c para globalizar las variables entre todos los ficheros.

5.1.12. ADC.c:

Contiene las funciones relacionadas con el control del convertidor analógico digital, entre otras funciones, “ADC_ch_enable” que habilita los canales comparadores del ADC interno del microcontrolador PIC y “read_ADC” que habilita la lectura de los valores de cada bloque de celdas (Va y Vb) y de la corriente medida desde el sensor efecto Hall.

5.1.13. ADC.h:

Contiene las constantes que se declaran globales para todo el programa. Además, también contiene las declaraciones de las funciones que se utilizan para la lectura y establecimiento de datos del fichero ADC,c para globalizar las variables entre todos los ficheros.

5.1.14. I2C.c:

Contiene todas las funciones relacionadas con la comunicación sobre la interfaz I2C

5.1.15. I2C.h:

Contiene las constantes, la definición de registros y declaraciones de las funciones relacionadas con el I2C.

5.1.16. Structs.h:

Contiene las variables que determinan las propiedades y establecen los límites de la batería que van a gestionar. Éstas están reunidas en estructuras para diferenciarlas fácilmente de las variables internas del programa.

Existen tres estructuras:

```
typedef struct feedback {
    int integer;
    int decimal;
    char feedback_fault;
} Tfeedback;

typedef struct cell {
    int rate_tension;
    int max_chg_tension;
    int max_dchg_tension;
    int sec_tension;
    double t_max;
} Tcell;

typedef struct battery {
    int n_used_blockA_cells;
    int n_used_blockB_cells;
    long int max_chg_current;
    long int min_chg_current;
    long int peak_chg_current;
    long int max_dchg_current;
    long int peak_dchg_current;
    double chg_balancing;
    double stb_balancing;
    Tcell SOA;
} Tbattery;
```

FIGURA 35: ESTRUCTURAS DE VARIABLES

5.2. MÁQUINA DE ESTADOS PRINCIPAL:

Este diagrama corresponde a la actitud y respuesta que va a llevar a cabo la rutina principal del microcontrolador. En él aparecen tres variables globales determinantes en el conjunto del programa: MA, DCHG_CHG y CHARGED.

La variable MA indica si la tarjeta de control es la maestra (MA =1) o no (MA = 0). Aunque en este proyecto solamente se va a programar la tarjeta maestra, cuando MA = 1.

El valor de la variable DCHG_CHG indica los modos de funcionamiento permitidos dentro de cada iteración de la rutina principal. Según la situación presente en la batería en el momento de la iteración, resultará del programa un número de 0 al 3, que indicará el estado que es necesario que alcance el dispositivo. Además, para que no pueda cambiar más de una vez de estado, un MOSFET en una iteración, se crea una función llamada “MOSFET_enable” la cual es llamada solamente al final de la rutina principal.

VARIABLE	DESCARGA	CARGA	DESCRIPCIÓN
DCHG_CHG = 0	NO	NO	No se permite nada (Dispositivo apagado)
DCHG_CHG = 1	Sí	NO	Sólo permite la descarga (Modo Descarga)
DCHG_CHG = 2	NO	Sí	Sólo permite la carga (Modo Carga)
DCHG_CHG = 3	Sí	Sí	Se permite todo (Modo Espera)

Tabla 14: Variable DCHG_CHG.

Por otro lado, la variable CHARGED, indica si la batería ha pasado por el “Modo descarga” después de la última carga. Con esto se impide que el BMS intente continuar cargando una vez la batería haya llegado a su tensión máxima de carga. Tanto el estado uno como el estado dos, se refieren a la carga de la batería como se explica en el párrafo

anterior, pero los estados dos, tres y cuatro surgen para saber por qué motivo la batería ha dejado de cargar.

Si CHARGED = 2, la carga ha sido completa y las celdas están equilibradas, eso se diagnostica cuando el cargador entra en todas las fases de carga y por tanto se ha ido reduciendo la corriente de carga de forma gradual.

Si CHARGED = 3, significa que durante el estado de carga y sin la batería completamente cargada, alguna de las celdas ha rebasado el límite máximo de tensión de su Zona de Operación Segura. A partir de ese momento y hasta que la celda vuelva a entrar en la Zona de Operación Segura, no se permitirá la carga de dicho sistema de baterías. Se podría decir que la batería está desequilibrada.

Si CHARGED = 4, significa que durante el estado de descarga y con la batería con suficiente tensión como para seguir funcionando, alguna de las celdas ha rebasado el límite mínimo de tensión de su Zona de Operación Segura. A partir de ese momento y hasta que la celda vuelva a entrar en la Zona de Operación Segura, no se permitirá la descarga de dicho sistema de baterías. Se podría decir que la batería está desequilibrada.

Los estados de la variable "CHARGED" son los siguientes:

- Si CHARGED = 0: Indica que la batería se ha descargado, aunque sea mínimamente después de la última carga.
- Si CHARGED = 1: La batería no se ha descargado, por tanto, no puede volver al "Modo Carga" o "CHG_mode".
- Si CHARGED = 2: La carga ha sido completa y se ha descargado en parte.

- Si CHARGED = 3: La carga se ha detenido antes de lo previsto porque hay, una celda como mínimo, que ha rebasado el límite de tensión máxima de carga. Por tanto, no permite la carga.
- Si CHARGED = 4: La descarga se ha detenido porque hay, una celda como mínimo, que ha sobrepasado el límite máximo de tensión de descarga.

VARIABLE	DESCRIPCIÓN DEL ESTADO	PERMISIBILIDAD
CHARGED = 0	La batería ha pasado por el modo descarga después de cargar.	Todo
CHARGED = 1	La batería no se ha descargado.	Descarga
CHARGED = 2	Batería cargada al completo.	Descarga
CHARGED = 3	Batería cargada pero con una celda habiendo rebasado el límite.	Descarga
CHARGED = 4	Batería descargada pero una celda ha rebasado el límite.	Carga

Tabla 15: Variable CHARGED.

Diagrama de flujo de la rutina principal

Marcos Ahulló Carbonell | July, 17th

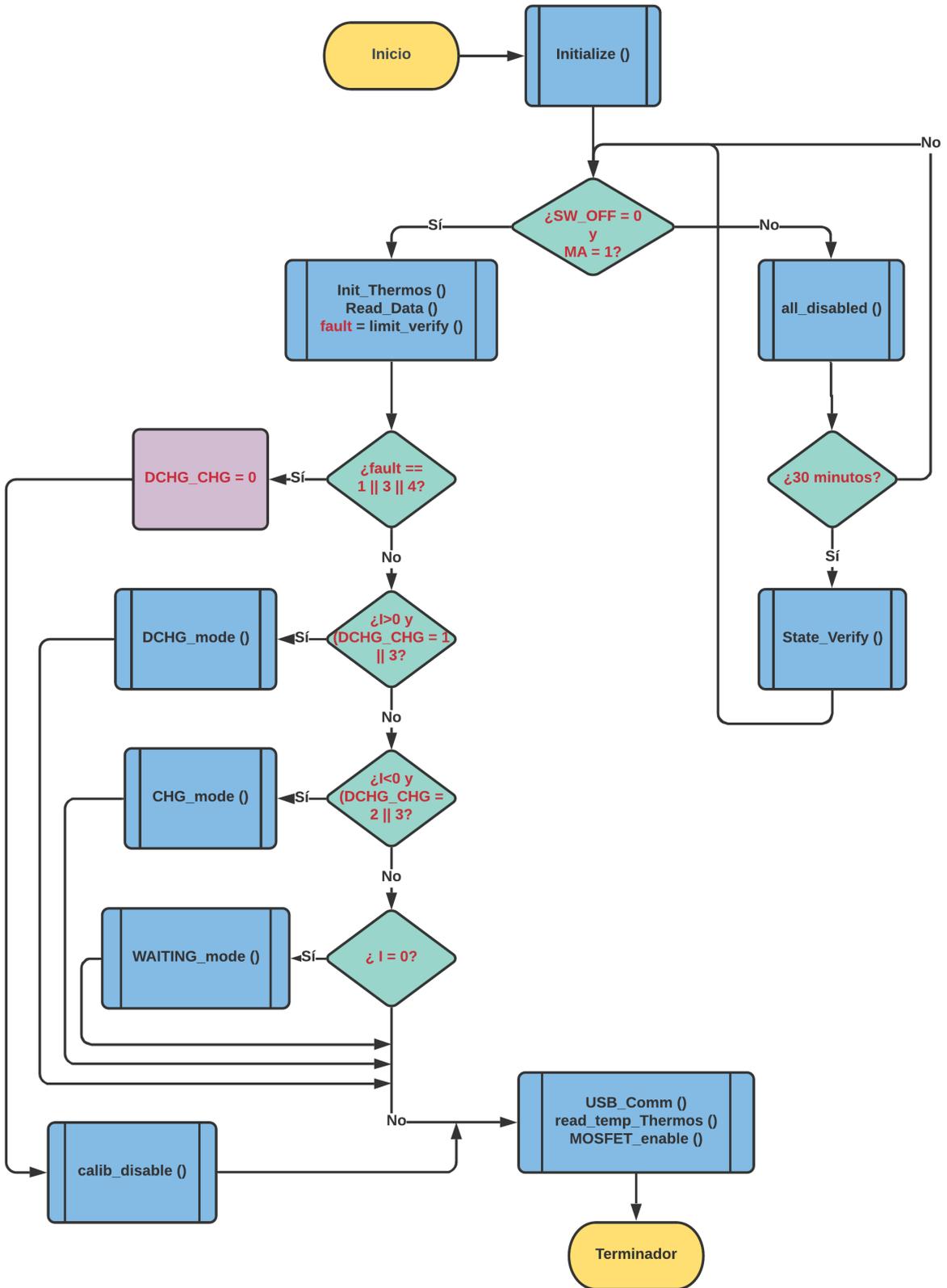


Tabla 16: Diagrama de flujo de la rutina principal

5.3. DESIGNACIÓN DE ENTRADAS Y SALIDAS:

El PIC seleccionado dispone de 100 pines correspondientes a entradas y salidas. En este apartado se define la función y conexión de cada pin.

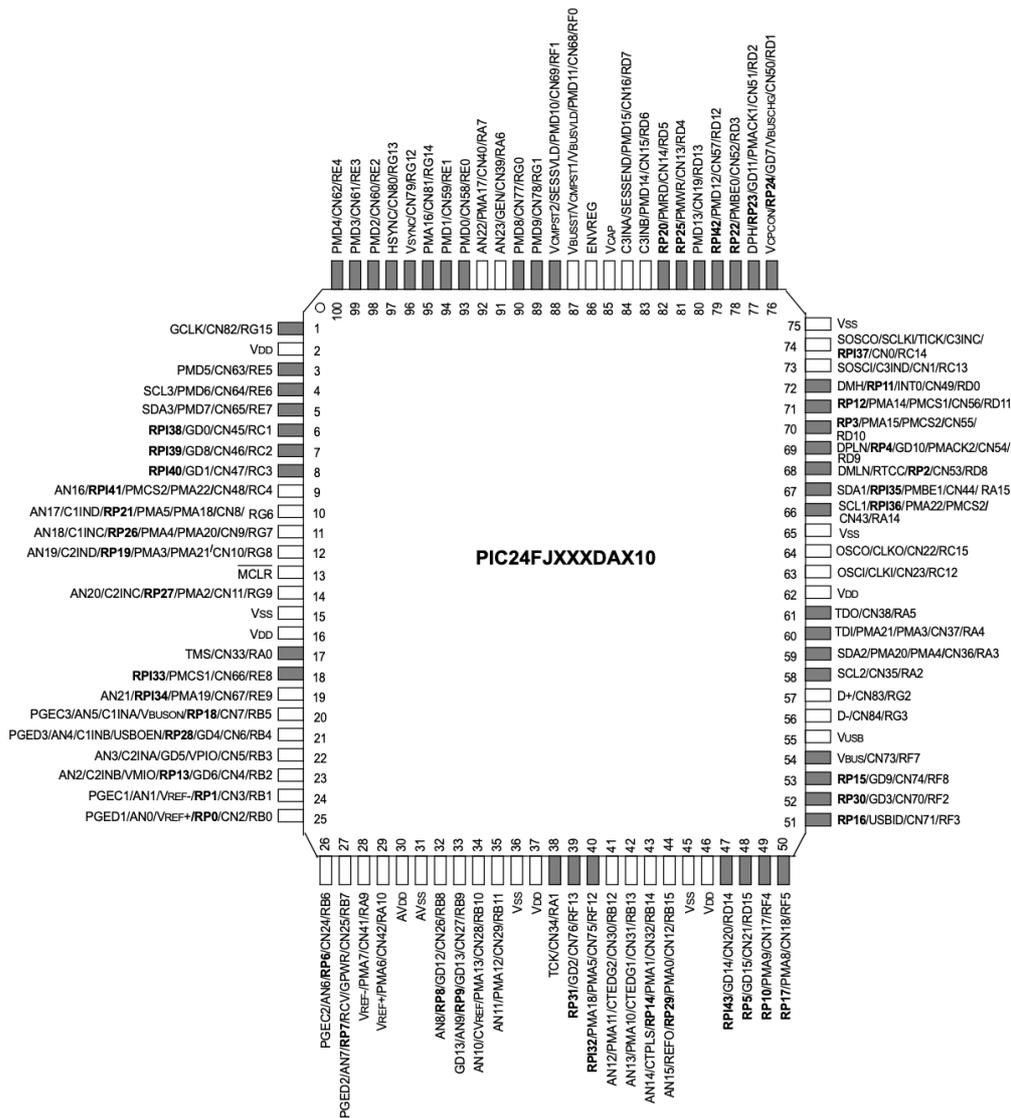
Nº PIN	NOMBRE	CONEXIÓN	DESCRIPCIÓN
1			No conectado
2	PWR_SPLY	VDD	3,3V Power Supply
3	DCHG_OUT	RE5 (Output)	Control del MOSFET de carga
4	CHG_OUT	RE6 (Output)	Control del MOSFET de descarga
5	EN_FAN	RE7 (Output)	Control del MOSFET del ventilador
6	EMER_LED	RG6 (Output)	Control del indicador de fallo
7			No conectado
8			No conectado
9			No conectado
10			No conectado
11			No conectado
12	RX	RG8 (Input)	Recepción protocolo comunicación USB
13	EN_CALIB	MCLR (Input)	Reseteo del controlador
14	TX	RG9 (Output)	Transmisión protocolo comunicación USB
15	GND	VSS	Ground
16	PWR_SPLY	VDD	3,3V Power Supply
17			No conectado
18			No conectado
19			No conectado
20	T4	RB5 (Output)	Encendido del sensor 4 (Ambiente)
21	T3	RB4 (Output)	Encendido del sensor 3 (Bloque B)
22	T2	RB3 (Output)	Encendido del sensor 2 (Bloque A)
23	T1	RB2 (Output)	Encendido del sensor 1 (Potencia)
24			No conectado
25	MA_SL1	RB0 (Input)	Master = 1 // Slave = 0

26			No conectado
27			No conectado
28			No conectado
29			No conectado
30	PWR_SPLY	AVDD	3,3V Power Supply
31	GND	AVSS	Ground
32	OFF_ON	RB8 (Input)	Interruptor de ON y OFF
33	MA_SL2	RB9 (Output)	Valida el estado de balanceo
34	EN_CELL	RB10 (Output)	Apaga el balanceo
35	A1_CELL	RB11 (Output)	Primera variable de elección de celdas
36	GND	VSS	Ground
37	PWR_SPLY	VDD	3,3V Power Supply
38			No conectado
39			No conectado
40			No conectado
41	A2_CELL	RB12 (Output)	Segunda variable de elección de celdas
42	A3_CELL	RB13 (Output)	Tercera variable de elección de celdas
43	A4_CELL	RB14 (Output)	Cuarta variable de selección de celdas
44	EN_CALIB	RB15 (Output)	Variable de activación del calibrado
45	GND	VSS	Ground
46	PWR_SPLY	VDD	3,3V Power Supply
47			No conectado
48			No conectado
49	SDA_EN	RF4 (Output)	SDA y parte baja de la barra LED
50	SCL_EN	RF5 (Output)	SCL y parte alta de la barra LED
51	CLK_ADC	RF3 (Output)	Clock del ADC
52	CELL15_PIC	RF2 (Input)	Lectura celda 14 (ADC)
53	CELL16_PIC	RF8 (Input)	Lectura celda 15 (ADC)
54	RESET_C	RF7 (Output)	Reset de celdas
55	VUSB	VSS	USB Ground
56	A1_MUX	RG3 (Output)	Primera variable de selección del canal multiplexor
57	A0_MUX	RG2 (Output)	Segunda variable de selección del canal multiplexor

58			No conectado
59			No conectado
60			No conectado
61			No conectado
62	PWR_SPLY	VDD	3,3V Power Supply
63			No conectado
64	A2_MUX	RC15 (Output)	Tercera variable de selección del canal multiplexor
65	GND	VSS	Ground
66			No conectado
67			No conectado
68	EN_MUX	RD8 (Output)	Habilitación de los multiplexores
69	CLK	RD9 (Output)	Reloj del microcontrolador
70	CONVST	RD10 (Output)	Señal de seguimiento e inicio de conversión
71	WR_ADC	RD11 (Output)	Señal de escritura del ADC
72	RD_ADC	RD0 (Output)	Habilitación lectura ADC
73	EOLC	RC13 (Input)	Señal de fin de la última conversión
74	EOC	RC14 (Input)	Señal de fin de conversión
75	GND	VSS	Ground
76	CELL1_PIC	RD1 (Input)	Lectura celda 0 (ADC)
77	CELL2_PIC	RD2 (Input)	Lectura celda 1 (ADC)
78	CELL3_PIC	RD3 (Input)	Lectura celda 2 (ADC)
79	Vo_B	RD13 (Input)	Señal de medida de tensión celdas bloque B
80	Vo_A	RD12 (Input)	Señal de medida de tensión celdas bloque A
81	CELL4_PIC	RD4 (Input)	Lectura celda 3 (ADC)
82	CELL5_PIC	RD5 (Input)	Lectura celda 4 (ADC)
83	CELL6_PIC	RD6 (Input)	Lectura celda 5 (ADC)
84	CELL7_PIC	RD7 (Input)	Lectura celda 6 (ADC)
85	GND	VCAP	External filter capacitor connection
86	PWR_SPLY	ENVREG	3,3V Voltage Regulator Enable
87	CELL8_PIC	RF0 (Input)	Lectura celda 7 (ADC)
88	CELL9_PIC	RF1 (Input)	Lectura celda 8 (ADC)
89			No conectado

90			No conectado
91			No conectado
92			No conectado
93	CELL10_PIC	RE0 (Input)	Lectura celda 9 (ADC)
94	CELL11_PIC	RE1 (Input)	Lectura celda 10 (ADC)
95	CUR_PIC	RG14 (Input)	Señal de medida de corriente
96			No conectado
97			No conectado
98	CELL_11	RE2 (Input)	Lectura celda 11 (ADC)
99	CELL_12	RE3 (Input)	Lectura celda 12 (ADC)
100	CELL_13	RE4 (Input)	Lectura celda 13 (ADC)

Tabla 17: Definición de los pines del PIC24FJ256DA210.



5.4. FUNCIONES PRINCIPALES:

5.4.1. Funciones relacionadas con la inicialización:

Este fichero contiene tres funciones diferentes encargadas de las inicializaciones generales. Las funciones son: *Initialize*, *config_ports* y *config_oscil*.

La función *Initialize* inicializa todas las variables globales del programa, las señales del BMS y calibra el sensor de corriente de efecto Hall. Este sensor tiene un offset de medida que va cambiando gradualmente con el tiempo, por tanto, siempre que los MOSFETS están en corte o se produce la inicialización, una parte de la función se dedica a calibrar ese sensor y ajustar el valor “offset” obtenido en el programa.

Las subrutinas a las que llama esta función para llevar a cabo la inicialización completa y la configuración de los dispositivos correspondientes son: *config_UART*, *I2C_init*, *MOSFET_enable*, *LED_stripe*, *battery.info*, *calib_disable*, *ADC_enabled*, *read_data*.

La función *config_ports* configura el modo de funcionamiento de todos los pines, según los pines que se conecten en el microcontrolador y se programen en el script, se necesitará definir un carácter u otro.

```
ANSBbits.ANSB5 = DIGITAL;
TRISBbits.TRISB5 = OUTPUT; // RB5 (T1)
ODCBbits.ODB5 = OPEN_DRAIN;

ANSBbits.ANSB4 = DIGITAL;
TRISBbits.TRISB4 = OUTPUT; // RB4 (T2)
ODCBbits.ODB4 = OPEN_DRAIN;

ANSBbits.ANSB3 = DIGITAL;
TRISBbits.TRISB3 = OUTPUT; // RB3 (T3)
ODCBbits.ODB3 = OPEN_DRAIN;

ANSBbits.ANSB2 = DIGITAL;
TRISBbits.TRISB2 = OUTPUT; // RB2 (T4)
ODCBbits.ODB2 = OPEN_DRAIN;
```

```
ANSFbits.ANSF0 = DIGITAL;
TRISFbits.TRISF0 = INPUT; // RF0
TRISFbits.TRISF1 = INPUT; // RF1
TRISFbits.TRISF3 = OUTPUT; // RF3
TRISFbits.TRISF7 = OUTPUT; // RF7

TRISEbits.TRISE0 = INPUT; // RE0
TRISEbits.TRISE1 = INPUT; // RE1
TRISEbits.TRISE2 = INPUT; // RE2
TRISEbits.TRISE3 = INPUT; // RE3
TRISEbits.TRISE4 = INPUT; // RE4
```

FIGURA 36: DEFINICIÓN DE LAS SEÑALES CONECTADAS

La función *config_oscil* selecciona y configura las propiedades del oscilador. En este caso, se utiliza el Oscilador Interno Primario y se ajusta por medio de la configuración siguiente:

- 001= Fast RC Oscillator with Postscaler and PLL module (FRCPLL).

Mediante el Postscaler y el PLL module se ajusta la frecuencia del oscilador a 32MHz.

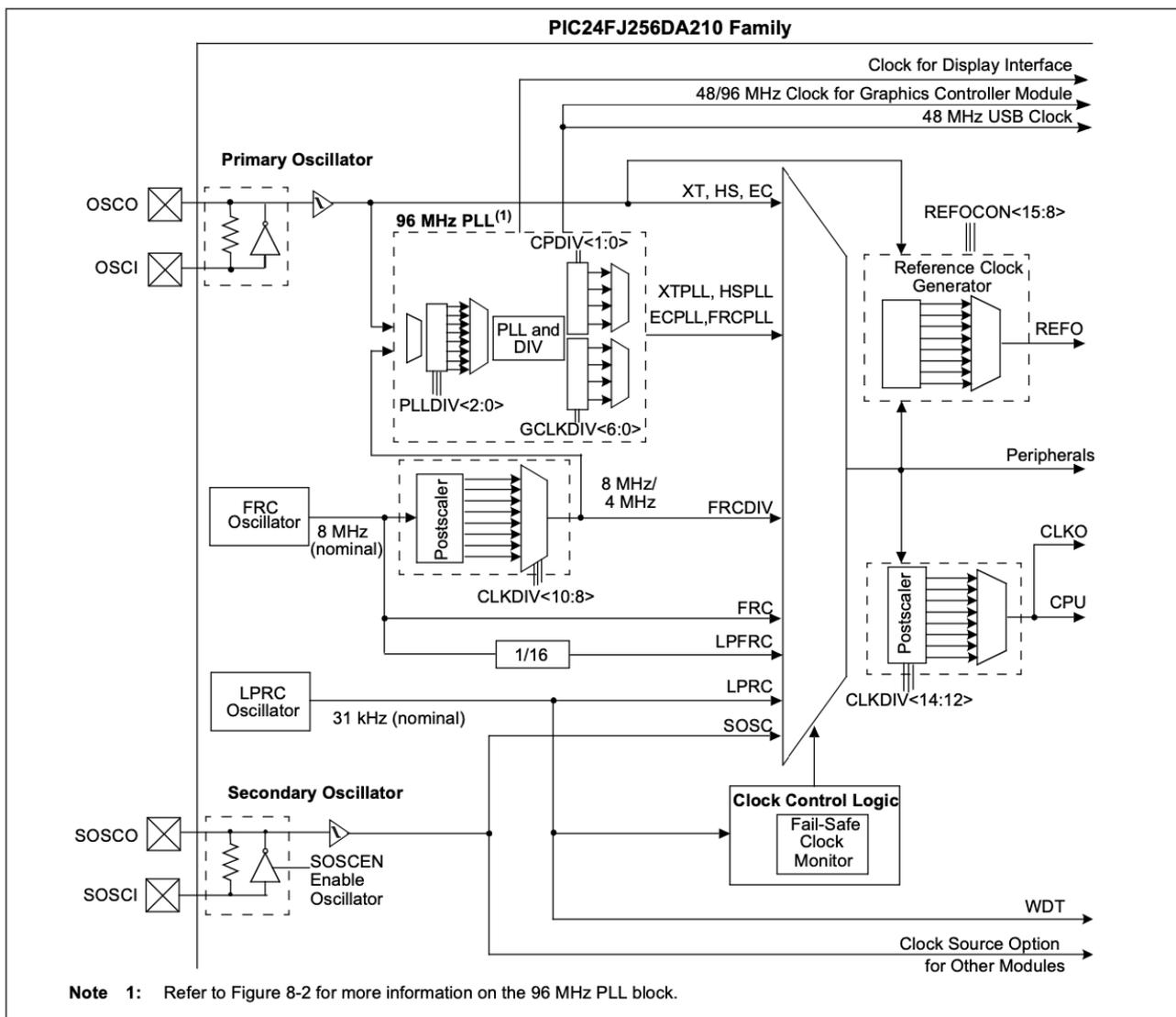


FIGURA 37: OSCILADORES EXISTENTES EN EL PIC

Otro fichero relacionado con la inicialización es *auxiliar.h*, en esta función se declaran las funciones utilizadas en el *main.c* y las constantes para todo el programa.

En el fichero *Structs.h* se declara un nuevo tipo de variable de estructura, cuyo objetivo es guardar toda la parametrización de configuración de la batería, Esta variable consiste en una estructura y tiene los siguientes campos:

```
typedef struct battery {
    int block_A_cells;           // Número de celdas conectadas en el bloque A
    int block_B_cells;           // Número de celdas conectadas en el bloque B
    long int max_chg_current;     // Corriente máxima de carga
    long int min_chg_current;     // Corriente mínima de carga
    long int peak_chg_current;    // Corriente máxima de pico de carga
    long int max_dchg_current;    // Corriente máxima de descarga
    long int peak_dchg_current;   // Corriente de pico máxima de descarga
    double chg_balancing;        // Porcentaje de balanceo en carga
    double stb_balancing;        // Porcentaje de balanceo durante estabilización
    Tcell SOA;                   // Datos SOA de la batería
} Tbattery;
```

FIGURA 38: CAMPOS EXISTENTES EN LA VARIABLE BATTERY.INFO

En la cual se puede escribir el valor adecuado en cada una de las variables desde el script de *Initialize*.

```
//Inicialización de battery_info por defecto
battery_info.n_used_blockA_cells = 5;
battery_info.n_used_blockB_cells = 5;
battery_info.max_chg_current = -10000;
battery_info.min_chg_current = -500;
battery_info.peak_chg_current = -20000;
battery_info.max_dchg_current = 35000;
battery_info.peak_dchg_current = 50000;
battery_info.chg_balancing = 1.007;
battery_info.stb_balancing = 1.005;
battery_info.soa.t_max = 55.0;
battery_info.soa.sec_tension = 3650;
battery_info.soa.max_chg_tension = 3600;
battery_info.soa.Vrate = 3200;
battery_info.soa.max_dchg_tension = 2500;
```

FIGURA 39: VALORES DE LOS CAMPOS DE BATTERY.INFO

También existen dos estructuras más llamadas “Tfeedback” y “Tcell”.

5.4.2. Funciones relacionadas con los termómetros:

Para la lectura de los termómetros se usa una librería ya creada (*OneWire.c*) pero con nombre cambiado (*Thermos.c*) y con pequeñas modificaciones. Al estar cada termómetro conectado a una entrada diferente solo se usan las funciones: *low_Thermos_drive*, *high_Thermos_drive*, *read_Thermos*, *reset_Thermos*, *write_bit_Thermos*, *read_bit_Thermos*, *write_byte_Thermos* y *read_byte_Thermos*.

```
#define T1_direc TRISBbits.TRISB5
#define T1_write LATBbits.LATB5
#define T1_read PORTBbits.RB5
#define T1_open_drain ODCBbits.ODB5 = 1; // Open drain = 1

// Sensor de Temperatura 2
// Sensor de Temperatura 2

#define T1_direc TRISBbits.TRISB4
#define T1_write LATBbits.LATB4
#define T1_read PORTBbits.RB4
#define T1_open_drain ODCBbits.ODB4 = 1; // Open drain = 1

// Sensor de Temperatura 3
// Sensor de Temperatura 3

#define T1_direc TRISBbits.TRISB3
#define T1_write LATBbits.LATB3
#define T1_read PORTBbits.RB3
#define T1_open_drain ODCBbits.ODB3 = 1; // Open drain = 1

// Sensor de Temperatura 4
// Sensor de Temperatura 4

#define T1_direc TRISBbits.TRISB2
#define T1_write LATBbits.LATB2
#define T1_read PORTBbits.RB2
#define T1_open_drain ODCBbits.ODB2 = 1; // Open drain = 1
```

También se crea un *Thermos.h* para declarar sus funciones y sus constantes.

Se modifican todas estas funciones para incluir las 4 sondas y se crean dos nuevas: *Init_Thermos* y *read_temp_Thermos*. Estas dos nuevas funciones son las que se usan en la rutina principal (*main*) y llaman a las anteriores.

FIGURA 30: DEFINICIÓN DE LOS PUERTOS DE LOS TERMÓMETROS

La función *init_Thermos* es la función encargada de iniciar la lectura de los valores de los sensores de temperatura, y la función *read_temp_Thermos* es la que lleva a cabo la obtención de la medida. Cuanto más tiempo haya transcurrido entre el inicio de las medidas y la obtención de la misma, más resolución se conseguirá, por eso, en la rutina principal, la función *init_Thermos* se lanza al principio y la obtención de la temperatura, es decir, la función *read_temp_Thermos* se produce al final.

La lectura de los termómetros se mide en paralelo para que no se sumen los tiempos de conversión y se guardan en variables generales como T1, T2, T3 y T4. Además, si un termómetro no está conectado, el programa devuelve el valor 9999, para verificar si está o no conectado.

5.4.3. Funciones relacionadas con la comunicación I2C:

La función *Comm_I2C* es la función que aparece en la rutina principal (*main*) para llamar a las subrutinas pertinentes para llevar a cabo la medición, transmisión, recibimiento o verificación de cualquier aspecto de la comunicación I2C. Con todo esto, verifica si ha habido alguna petición de escritura o lectura a través del segundo puerto de comunicación I2C del PIC. En caso afirmativo, se llama a la función *Tresponse_data*, quien le devuelve un valor entero determinando qué dispositivo o servicio ha hecho dicha petición, pudiendo pasar así los datos solicitados. También hay que tener en cuenta que la función *Tresponse_data* es común para la comunicación USB e I2C, por tanto, los valores de devolución serán los mismos y es necesario ver la tabla siguiente para determinar quién realiza la petición.

CASO	INFORMACIÓN DEVUELTA	FUNCIÓN
a	Tensión de la celda 1 (en mV)	(MSB*100) + LSB
b	Tensión de la celda 2 (en mV)	(MSB*100) + LSB
c	Tensión de la celda 3 (en mV)	(MSB*100) + LSB
d	Tensión de la celda 4 (en mV)	(MSB*100) + LSB
e	Tensión de la celda 5 (en mV)	(MSB*100) + LSB
f	Tensión de la celda 6 (en mV)	(MSB*100) + LSB
g	Tensión de la celda 7 (en mV)	(MSB*100) + LSB
h	Tensión de la celda 8 (en mV)	(MSB*100) + LSB
i	Tensión de la celda 9 (en mV)	(MSB*100) + LSB
j	Tensión de la celda 10 (en mV)	(MSB*100) + LSB
k	Tensión de la celda 11 (en mV)	(MSB*100) + LSB
l	Tensión de la celda 12 (en mV)	(MSB*100) + LSB
m	Tensión de la celda 13 (en mV)	(MSB*100) + LSB
n	Tensión de la celda 14 (en mV)	(MSB*100) + LSB
o	Tensión de la celda 15 (en mV)	(MSB*100) + LSB
p	Tensión de la celda 16 (en mV)	(MSB*100) + LSB

I	Corriente total instantánea	MSB = Integer, LSB = decimal
C	Estado del calibrado	LSB = Primer bloque, MSB = segundo Cx = 0 apagado, Cx = 1 encendido
R	Sensor de temperatura 1	MSB = Parte entera
S	Sensor de temperatura 2	LSB = Parte decimal
T	Sensor de temperatura 3	9999 = No conectado
V	Sensor de temperatura 4	8500 = Termómetro roto
A	Tensión nominal de las celdas (en mV)	(MSB*100) + LSB
B	Tensión máxima de carga (en mV)	(MSB*100) + LSB
D	Tensión máxima de descarga (en mV)	(MSB*100) + LSB
E	Zona de operación segura (en mV)	(MSB*100) + LSB
F	Temperatura máxima	MSB = Integer, LSB = decimal
G	Número de celdas usadas	MSB = Bloque B, LSB = Bloque A
H	Corriente máxima de carga	MSB = Parte entera
J	Corriente mínima de carga	LSB = Parte decimal
K	Corriente en pico de carga	MSB = 12
L	Corriente máxima de descarga	LSB = 14
M	Corriente máxima de descarga	Corriente = 12,14 A
Default	Ninguno de los casos anteriores	Tresponse_data = 1

Tabla 18 : Casos posibles de la función de respuesta.

Los datos de respuesta a las peticiones se componen de dos bytes llamados *integer* y *decimal*. *I2C_comm* llama a la función *SlaveReadI2C*, una vez recibida la petición correcta (su dirección), para obtener el resto de los datos. Después, llama a la función *SlaveWriteI2C* para enviar los datos solicitados. Durante esta secuencia de transmisión, el BMS debe generar la señal ACK correspondiente, para ello se llama a la función *AckI2C*. El BMS genera la señal ACK tres veces en cada petición.

Por último, la función *I2C_init* es llamada desde la función *Initialize*, al arrancar el BMS, para configurar el segundo puerto I2C como esclavo a la frecuencia estándar de 1 KHz y

asignarle la dirección 0x28. La frecuencia de funcionamiento del I2C se selecciona calculando el valor apropiado para el registro I2C2BRG.

Para permitir la transmisión de bits se emplean las funciones siguientes:

- I2C_init:

```
void I2C_init(unsigned char F)
{
    int temp;

    I2C2BRG = F; // I2C speed
    I2C2CONbits.I2C2CEN = 0; // I2C communication disable = 0
    I2C2CONbits.DISSLW = 1; // Slew Rate control disable = 1
    IFS3bits.SI2C2IF = 0; // Clear Slave Interrupt Flag = 0
    IFS3bits.MI2C2IF = 0; // Clear Master Interrupt Flag = 0
    I2C2CONbits.IPMIEN = 1; // I2C module is operating as a Slave
    I2C2CONbits.I2CEN = 1; // Enable I2C module and configure pins
    temp = I2C2RCV; // Read buffer to clear buffer full
    IEC3bits.MI2C2IE = 0; // Disable Master I2C2 Event Interrupt
    IEC3bits.SI2C2IE = 0; // Disable Slave I2C2 Event Interrupt
    I2C2CONbits.SCLREL = 0; // Holds SCLx clock low (clock stretch)
    I2C2CONbits.A10M = 0; // I2CxADD is a 7-bit slave address = 0
    I2C2ADD = SLAVE_ADDRESS; // Address of slave
}
```

FIGURA 41: CARACTERÍSTICAS DEL I2C

- SlaveReadI2C:

```
unsigned char SlaveReadI2C (void)
{
    while(!I2C2STATbits.RBF); // Se mantiene hasta que se reciba algún dato
    I2C2STATbits.I2COV = 0;
    return (I2C2RCV);
}
```

FIGURA 42: FUNCIÓN PARA LEER BITS EN I2C.

- SlaveWriteI2C:

```
void SlaveWriteI2C (unsigned char data_out)
{
    I2C2TRN = data_out; // Datos enviados a I2C2TRN register
    I2C2CONbits.SCLREL = 1; // Release the clock
}
```

FIGURA 43: FUNCIÓN PARA ESCRIBIR BITS EN I2C.

- AckI2C:

```
void AckI2C(void)
{
    I2C2CONbits.ACKDT = 0;
    I2C2CONbits.ACKEN = 1;
}
```

FIGURA 44: FUNCIÓN DE RECONOCIMIENTO DE BITS EN I2C.

5.4.4. Funciones relacionadas con la comunicación USB:

Para enviar y recibir (escribir y leer) información usando el protocolo USB se emplean las funciones programadas en el fichero “Communications”: “USB_comm”, “RX”, “TX”, “Tresponse_data”, config_UART.

“Communications” es la función a la que se llama desde la función principal (“main.c”) y la que verifica si ha recibido algún dato. En caso afirmativo, consulta con la función “Tresponse_data” concretando cuál es la petición y qué datos se envían o reciben.

Los datos de “Tresponse_data” siempre están compuestos por dos bytes, la parte entera y la parte decimal (integer y decimal), “Communications” llama a la función *RX* para verificar si hay alguna petición de datos y a *TX* para enviar los datos solicitados.

La función “Tresponse_data” es común para los dos protocolos de comunicación, USB e I2C. Por tanto, la tabla 18 también enumera los casos en los que se envía algo por USB.

Cada tensión de las celdas se solicita con una letra (desde la a hasta la p) y la corriente con la letra l. Estos datos en el PIC se procesan en mV y mA respectivamente, para no trabajar con números decimales, son descompuestos en dos partes para enviarlos por el USB. La parte entera se multiplica por 100 para ser utilizada como las centenas, y la parte decimal se suma a la parte entera para resultar un número formado por tres o cuatro dígitos representando la tensión en mV.

El estado del balanceo se representa con una C. Dos bytes representan a esta variable, uno por bloque de celda, y cada bit propio representa una celda.

La lectura de los sensores de temperatura se representa con R, S, T y V, midiendo los valores del sensor número uno hasta el número cuatro respectivamente. Respetándose también la lectura de los dos bits, integer y decimal, para saber la parte entera y la parte decimal.

Por último, las mediciones de las celdas se solicitan con A, B, D y E. Y los parámetros de la batería se solicitan con H, J, K, L y M.

5.4.5. Funciones relacionadas con el balanceo:

El balanceo se controla con el fichero “Balancing.c”, el cual utiliza las siguientes funciones: “Cell_calib”, “calibrate” y “calib_disable”.

Las funciones “cell_calib” y “calib_disable” son dos funciones casi continuamente en funcionamiento por todo el programa. En contra, la función “calibrate” solo se usa cuando es llamada por “cell_calib” para activar o desactivar el balanceo de una celda en concreto.

La función “calib_disable”, conectada a la señal de reset, permite desactivar el balanceo de todas las celdas a la vez, siendo lo adecuado en el modo de descarga (DCHG_mode) y con la aparición de algún fallo.

Por contrario, en el modo de carga (CHG_mode) se llama constantemente a la función “cell_calib” con la variable “percent” concretada por el usuario en los datos de batería (battery_info). Se toma como referencia la celda de menor voltaje y se le aplica el

balanceo a todas aquellas celdas que tengan una tensión mayor a la tensión de la celda menos cargada multiplicado por el porcentaje escrito.

En este caso, si la celda de menor tensión tuviera 3320 mV, el BMS activaría el MOSFET y por tanto el balanceo de todas las celdas con una tensión superior a:

$$\mathbf{3320mV * 1,007 (\%) = 3343mV}$$

Se ha programado de manera que sea muy ágil y fácil modificar la variable “percent” para adaptarlo a cualquier tipo de celdas, dado que está declarada en la estructura de “Battery_info” y es una variable de entrada de la propia función “cell_calib”.

Una vez finalizada la carga y pasado al “WAITING_mode”, con la batería completamente cargada y ningún error, es decir, con la variable CHARGED = 2, también se activa el balanceo para equilibrar lo máximo posible las tensiones de cada celda, con esto, se podrá incrementar el rendimiento y la vida útil de dicha batería y garantizará un buen comportamiento independientemente de los ciclos de carga y descarga que se le apliquen. En este modo se recomienda un ajuste más estricto para obtener el máximo rendimiento posible, definiéndose así un porcentaje más pequeño, stb_balancing = 1,005.

Por tanto, si la celda con menos tensión tiene 3585mV, se activará el balanceo de todas las celdas con una tensión superior a:

$$\mathbf{3585mV * 1,005 (\%) = 3603mV}$$

Siendo el valor de 3,6V (3600 mV) la máxima tensión de carga admitida para las celdas LiFePO4 y determinada por la Zona de Operación Segura.

5.4.6. Funciones relacionadas con la verificación:

Se han definido dos funciones de verificación, ambas programadas en el mismo archivo que la rutina principal “main.c”. La primera se usa en la rutina principal cuando el BMS está apagado “state_verify” y la segunda se usa cuando el BMS está encendido “limit_verify”.

“state_verify” revisa los sistemas de medida, vuelve a calcular el offset del sensor de corriente efecto Hall y llama a la función “limit_verify” para que realice las comprobaciones necesarias de la temperatura, tensión y corriente para verificar que no hay ningún error.

Esta función devuelve un número entero concretando el estado de las verificaciones, esa parte entera responde a la siguiente tabla:

VARIABLE	DESCRIPCIÓN DEL ESTADO
Fault = 0	Sin errores, todo verificado.
Fault = 1	Error en alguno de los sensores de temperatura
Fault = 2	Tensión más baja que la mínima de descarga
Fault = 3	Tensión más alta que la máxima de carga
Fault = 4	Error de corriente
Fault = 5	Celdas incorrectas

Tabla 19: Estados del valor devuelto “fault”.

“limit_verify” también puede encender el MOSFET que controla el ventilador, si se presenta el caso que la temperatura es más alta que el límite definido en la estructura de “battery_info” (T_Fan = estándar 50.0°C). En este caso también se ha programado esta variable como una variable de entrada a la función, por tanto, presenta la misma facilidad de modificación que los porcentajes de balanceo.

Esta función también controla el LED de emergencia, que se enciende cuando la batería está baja y es necesario conectarla al cargador (estando tanto encendida como apagada), o bien cuando una celda como mínimo sobrepasa el límite de tensión máxima o mínima, habilitando en cada caso correspondiente el modo carga o descarga.

5.4.7. Funciones relacionadas con la barra de LED's:

Esta función está implementada en el fichero "main.c", y controla la barra de LED's que representa la medición de batería.

Esta función tiene tres estados: apagado, indicador de carga e indicador SOC. Según el valor de la variable de entrada "activate", de carácter entero corto, la barra permanece apagada (ACTIVATE = 0) y estado apagado, la barra se ilumina secuencialmente para indicar el modo carga (ACTIVATE = 1) con el estado indicador de carga. Y además, en el modo descarga o espera, la batería indica el nivel de carga actual (ACTIVATE = 2). Los led's rojos indican el nivel de batería más bajo, los led's verdes de la zona indican un nivel de batería intermedio y los led's verdes indican que la batería a plena carga.

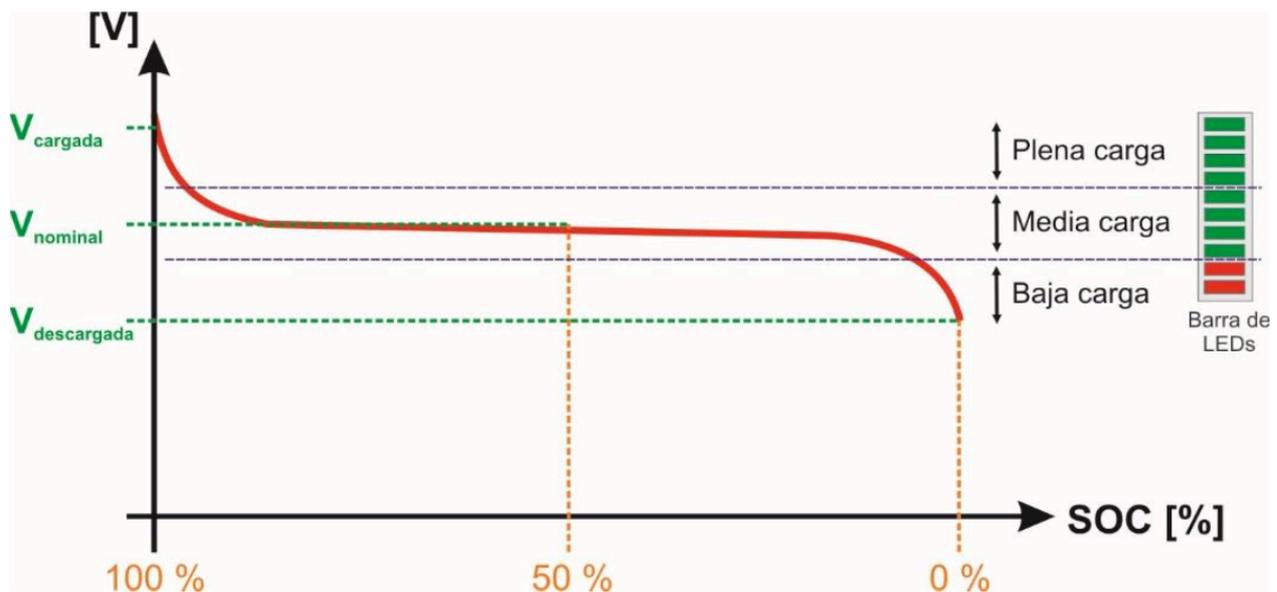


FIGURA 45: BARRA DE LEDS RESPECTO AL NIVEL DE BATERÍA.

Normalmente el medidor de corriente se implementa con un integrador de corriente, es decir, sumando el consumo instantáneo en el caso de la descarga y restando la intensidad de carga instantánea en el caso de la carga teniendo una base de tiempo real. Pero, por una implementación más rápida y sin necesidad de encontrar una precisión alta en la medida para la presente barra, se toma como referencia la tensión nominal de la batería y se establecen las tres zonas. En función de la zona en la que se encuentre la tensión actual de la batería se iluminan más o menos LEDs de la barra.

5.4.8. Funciones relacionadas con el convertidor ADC.

Para llevar a cabo la lectura de las tensiones de cada celda por los canales comparadores del ADC interno del microcontrolador PIC, se emplean las funciones implementadas en el archivo "ADC.c" como por ejemplo: "read_ADC" y "ADC_ch_enable". Como también, alguna de las funciones implementadas en el fichero "reads&mux.c" como "channel_sel_mux".

La función usada en la rutina principal con la que se accede a estos datos es "read_data", la cual devuelve los valores de las tensiones de cada celda, la tensión total de la batería multiplicando el voltaje de cada una por el número total de celdas existentes y conectadas, y la corriente de salida.

"read_data" llama a la función "channel_sel_MUX" para seleccionar por medio de los bucles, todas las celdas de la batería. Cuando se elige un canal del multiplexor, tras el tiempo de estabilización, se llama a la función "read_ADC" para obtener n muestras de cada canal. El valor de n se encuentra almacenado en una constante llamada "average", para que pueda ser modificado con facilidad. Por defecto, "average=50 muestras". Se realiza este sobremuestreo para mejorar la resolución de la conversión.

6. PRESUPUESTO:

En el presupuesto se van a acumular los precios y cantidades de cada uno de los componentes, materiales para la fabricación de la placa y horas de trabajo del ingeniero.

6.1. COMPONENTES ELECTRÓNICOS:

Componente	Precio Unidad	Nº Unidades	Precio Total
TLP240GA	1,49 €	16	23,84 €
MUX507IPWR	3,15 €	2	6,30 €
AD8276ARMZ-R7	2,62 €	4	10,48 €
ADUM1400BRWZ	5,44 €	2	10,88 €
ADUM3210TRZ-EP	6,28 €	2	12,56 €
REF3330AIDBZT	1,83 €	2	3,66 €
OPA2141AID	3,29 €	6	19,74 €
ACPL-C87A-500E	4,64 €	6	27,84 €
RPMB5.0-2.0	5,73 €	1	5,73 €
MGXS6243R3	28,76 €	1	28,76 €
PIC24FJ256D210	6,53 €	1	6,53 €
MAX31820PARMCR	1,96 €	4	7,84 €
CP2102N-A02-GQFN20	1,20 €	1	1,20 €
BS170	0,405 €	6	1,62 €
STP130N8F7	1,53 €	2	3,06 €
MUR.460	0,20 €	2	0,40 €
ISH3.315A	6,75 €	2	13,5 €
LAM 3 K 100 05	35,64 €	1	35,64 €
ACS781LLRTR-100B-T	4,86 €	1	4,86 €
Componentes pasivos	10,00 €	-	10,00 €
Conectores	5,00 €	-	5,00 €
Total			239,44 €
Total + I.V.A. (21%)	50,28 €		289,72 €

6.2. MATERIALES DE FABRICACIÓN:

Material	Precio lote	Precio material necesario
Board fibra de vidrio (100mm*160mm)	7,35 €	7,35 €
Partes metálicas (tornillos, arandelas, etc.)	7,00 €	7,00 €
Lámina de cobre	3,40 €	3,40 €
Total		17,75 €
Total + I.V.A. (21%)	3,72 €	21,47 €

6.3. HORAS DE DISEÑO:

Carácter	Precio hora	Horas totales	Precio total
Ingeniero eléctrico	45,00 €	40	1800 €
Ingeniero software	45,00 €	40	1800 €
Total		80	3600 €
Total + I.V.A. (21%)	756 €		4356 €

6.4. PRECIO TOTAL DEL PROYECTO:

Apartado	Precio sin I.V.A.	Precio con I.V.A.
Componentes electrónicos	239,44 €	289,72 €
Materiales de fabricación	17,75 €	21,47 €
Horas de diseño	3600 €	4356 €
Total	3850,66 €	4667,20 €

7. CONCLUSIONES:

A lo largo de este Trabajo Fin de Máster se ha desarrollado el diseño de un BMS para baterías de tecnología LiFePO₄ cumpliendo con éxito todos los objetivos propuestos. Tanto la metodología como la planificación se han seguido sin problema, obteniéndose los resultados esperados.

A la vista de las simulaciones se denota la importancia del BMS en las baterías LiFePO₄, no solo para garantizar la seguridad, sino también, para obtener un mayor rendimiento y alargar su vida útil junto con la cantidad de ciclos de carga/descarga.

Con este dispositivo se puede proteger cualquier sistema de baterías con tecnología LiFePO₄ para que sus celdas trabajen en todo momento, desde una batería de cuatro celdas, a una batería de sesenta y cuatro celdas, pudiendo ajustar los parámetros de dicha batería en el microcontrolador PIC fácilmente.

Con las simulaciones, también se demuestra la importancia del balanceo en el modo de carga, gracias a esto se puede garantizar un equilibrio de todas las celdas durante toda su vida útil.

El único inconveniente puede ser el precio del BMS como producto, puede llegar a alrededor de los 400€ si se incluye el gasto de la fabricación de las placas y eso puede hacer aumentar demasiado el precio de las baterías, sobre todo de las pequeñas. Siendo optimizable con algunos trabajos futuros mostrados en la página siguiente.

7.1. TRABAJOS FUTUROS:

- Cambiar el método de medición de corriente para que sea con integración en vez de medición instantánea para obtener más precisión en las medidas de corriente y poder calcular en todo momento el porcentaje de batería restante del sistema.
- Diseñar las placas esclavas.
- Eliminar el controlador CP2102N-A02-GQFN20 y conectar la comunicación USB al puerto existente en el propio PIC.
- Diseñar las placas de circuito impreso que componen el BMS.
- Fabricar las placas de circuito impreso que componen el BMS.
- Implementar un protocolo de comunicaciones con mayor robustez. Modbus o Ethernet/IP.
- Diseñar el programa de las placas esclavo para tener la posibilidad de ampliar los módulos del BMS si alguna batería lo requiere.
- Diseñar una interfaz I2C para que el usuario pueda controlar y cambiar todas las variables necesarias según la distribución de las celdas en el sistema de baterías.
- Implementar todas las funciones de ahorro de batería que tengan los componentes.
- Crear un controlador de ROS para poder comunicarse con los PC's de las plataformas móviles de Robotnik.
- Mejorar la comunicación USB.

PLIEGO DE CONDICIONES



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

1. INTRODUCCIÓN:

El objetivo del presente pliego de condiciones es recopilar las características de cada uno de los materiales y componentes utilizados, ver que estos cumplen con la normativa correspondiente y explicar una serie de directivas básicas para el correcto funcionamiento del dispositivo encargado de la gestión de sistemas de batería LiFePO₄. Además, durante la realización del proyecto se ha prestado especial atención al marco legal en el cual se engloba el proyecto.

2. CONDICIONES GENERALES:

En lo relativo al marco legal, este proyecto se rige por la Normativa de Marco de Trabajos De Fin de Grado y Fin de Máster de la Universidad Politécnica de Valencia, aprobada en Consejo de Gobierno el 13 de marzo de 2018.

Esta normativa nace a consecuencia del Real Decreto 1393/2007, por el que se establece la ordenación de las enseñanzas universitarias oficiales, modificado por el Real Decreto 861/2010 que dispone, con carácter general, que todos los títulos oficiales “concluirán con la elaboración y defensa” de un Trabajo Fin de Grado (TFG) o Trabajo Fin de Máster (TFM), según el caso. Adicionalmente las órdenes ministeriales por las que se establecen los requisitos que deben cumplirse para la verificación de los títulos oficiales que habilitan para las profesiones de Arquitecto, Arquitecto Técnico, Ingeniero o Ingeniero Técnico también incluyen prescripciones generales relativas a la naturaleza de los TFG, condiciones para su presentación y defensa y, en algún caso, composición del tribunal calificador.

3. ESPECIFICACIONES TÉCNICAS:

3.1. COMPONENTES ANALOG DEVICES:

Los componentes escogidos en este proyecto del fabricante Analog Devices son:

- Aislador digital integrado en el circuito de medidas: ADUM1400BRWZ.
- Amplificador diferencial integrado en el circuito de medidas: AD8276ARMZ.
- Aislador digital integrado en el circuito de la placa de potencia: ADUM3210TRZ-EP.

Estos tres componentes cumplen con la totalidad de condiciones técnicas, los certificados de calidad que ofrece el fabricante Analog Devices son IATF 16949, ISO 9001, ISO 45001, ISO AS9100 y QML.

Además, este fabricante tiene un programa de monitoreo y predicción de fiabilidad muy activo para garantizar que todos los productos sean de la más alta calidad. Realiza todas las clases principales de pruebas de fiabilidad en cada uno de sus procesos utilizando equipos y metodologías de última generación. Los resultados de las pruebas aceleradas de estrés ambiental se extrapolan en condiciones de operación estándar para predecir vidas útiles y garantizar que sus productos tengan algunos de los niveles de fiabilidad más altos de la industria.



FIGURA 46: AISLADORES Y AMPLIFICADORES ANALOG DEVICES.



AHEAD OF WHAT'S POSSIBLE™

October 5, 2018

Re: Directive (EU) 2015/863 RoHS General Statement

To Whom It May Concern:

Analog Devices Inc. (ADI) lead (Pb) free part(s) and its homogeneous sub-components, to the Company's knowledge and after good faith inquiry, are certified not to contain lead, mercury, cadmium, hexavalent chromium, polybrominated biphenyls ("PBB"), polybrominated diphenyl ethers ("PBDE"), bis(2-ethylhexyl) phthalate ("DEHP"), butyl benzyl phthalate ("BBP"), dibutyl phthalate ("DBP") and diisobutyl phthalate ("DIBP") above the indicated levels set forth in the current Commission Delegated Directive (EU) 2015/863 of 31 March 2015, amending Annex II to Directive 2011/65/EU.

The information on ADI products in this letter is based upon the information provided by the material suppliers and assembly manufacturers used by ADI to manufacture these products. As such, ADI makes no independent representations or warranties, expressed or implied, and assumes no liability in connection with the use of this information.

Thank you for your interest in ADI products.

Susan Capuli
Director of Environmental, Health and Safety

3.2. COMPONENTES TEXAS INSTRUMENTS:

Los componentes escogidos para este proyecto del fabricante Texas Instruments son los siguientes:

- Multiplexor integrado en el circuito de medidas: MUX507IPWR.
- Referencia de tensión integrada en el circuito de acondicionamiento: REF3330AIDBZT.
- Amplificador operacional integrado en el circuito de acondicionamiento: OPA2141AID.

El fabricante Texas Instruments obtuvo por primera vez las Certificaciones del Sistema de Gestión de Calidad de la Organización Internacional de Normalización (ISO) (ISO 9001) y del Sistema de Gestión Ambiental (ISO 14001) en 1996 y desde entonces ha mantenido el cumplimiento de los requisitos de ISO. También están certificados con los siguientes estándares:

IATF 16949 certified in 2018 (Global automotive industry).

OHSAS 18001 certification in 2007 (Occupational health and safety).

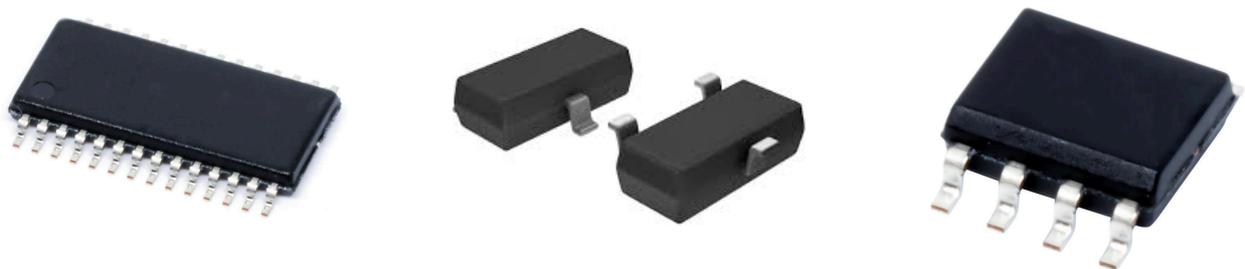


FIGURA 47: COMPONENTES TEXAS INSTRUMENTS

3.3. COMPONENTES COSEL:

El componente escogido para este proyecto del fabricante Cosel es el siguiente:

- Convertidor DC/DC no aislado con salida de 5V: MGXS6243R3.

Este fabricante cumple con los certificados ISO-9001 y ISO-14001. También cumple además con los certificados de medio ambiente, como son el RoHS y el REACH, siendo los dos certificados restrictivos para el uso de ciertos elementos químicos dentro de los diferentes productos fabricados.



FIGURA 48: APROBACIONES DE LOS CERTIFICADOS ROHS Y REACH.



FIGURA 49: CONVERTIDOR DC/DC COSEL

3.4. COMPONENTES SILICON LABS:

El componente escogido para este proyecto del fabricante Cosel es el siguiente:

- El módulo USB integrado en el circuito de comunicaciones: CP2102N-A02-GQFN20.

Este fabricante entró en el Sistema de gestión de calidad ISO 9001 en junio del 2000; en el sistema de gestión ambiental ISO 14001 en diciembre de 2006 y está comprometido con la excelencia en calidad y desempeño ambiental. Cada producto pasa por extensas pruebas de calificación antes del lanzamiento de producción. Silicon Labs califica los productos de circuito integrado (IC) siguiendo JEDEC JESD47, calificación de prueba de estrés de circuitos integrados.

Los productos automotrices de Silicon Labs cumplen con la calificación AEC-Q100 y solo utilizan proveedores certificados IATF 16949. Las ofertas incluyen productos de Grado 1, 2 y 3 que cumplen y exceden AEC-Q100, JEDEC y la mayoría de las especificaciones OEM, incluidas EMC y ESD.



FIGURA 50: MÓDULO DE COMUNICACIONES USB.

3.5. COMPONENTES MAXIM INTEGRATED:

El componente escogido para este proyecto del fabricante Maxim Integrated es el siguiente:

- Sensor de temperatura: MAX31820PARMCR.

El componente cumple con la totalidad de condiciones técnicas, los certificados de calidad que ofrece el fabricante Maxim Integrated son IATF 16949, ISO 9001, ISO 45001, ISO AS9100 y QML.



FIGURA 51: SENSOR DE TEMPERATURA.

3.6. COMPONENTES MICROCHIP:

El componente escogido para este proyecto del fabricante Microchip es el siguiente:

- Microcontrolador PIC: PIC24FJ256DA210 con 100 pines disponibles.

Features	PIC24FJ128DA110	PIC24FJ256DA110	PIC24FJ128DA210	PIC24FJ256DA210
Operating Frequency	DC – 32 MHz			
Program Memory (bytes)	128K	256K	128K	256K
Program Memory (instructions)	44,032	87,552	44,032	87,552
Data Memory (bytes)	24K		96K	
Interrupt Sources (soft vectors/NMI traps)	66 (62/4)			
I/O Ports	Ports A, B, C, D, E, F, G			
Total I/O Pins	84			
Remappable Pins	44 (32 I/O, 12 input only)			
Timers:				
Total Number (16-bit)	5 ⁽¹⁾			
32-Bit (from paired 16-bit timers)	2			
Input Capture Channels	9 ⁽¹⁾			
Output Compare/PWM Channels	9 ⁽¹⁾			
Input Change Notification Interrupt	84			
Serial Communications:				
UART	4 ⁽¹⁾			
SPI (3-wire/4-wire)	3 ⁽¹⁾			
I ² C™	3			
Parallel Communications (EPMP/PSP)	Yes			
JTAG Boundary Scan	Yes			
10-Bit Analog-to-Digital Converter (ADC) Module (input channels)	24			
Analog Comparators	3			
CTMU Interface	Yes			
USB OTG	Yes			
Graphics Controller	Yes			
Resets (and delays)	POR, BOR, RESET Instruction, MCLR, WDT; Illegal Opcode, REPEAT Instruction, Hardware Traps, Configuration Word Mismatch (OST, PLL Lock)			
Instruction Set	76 Base Instructions, Multiple Addressing Mode Variations			
Packages	100-Pin TQFP and 121-Pin BGA			

El microcontrolador cumple con todas las condiciones técnicas específicas para poder integrarse en una placa de circuitos integrados y mantiene todas las garantías del fabricante.

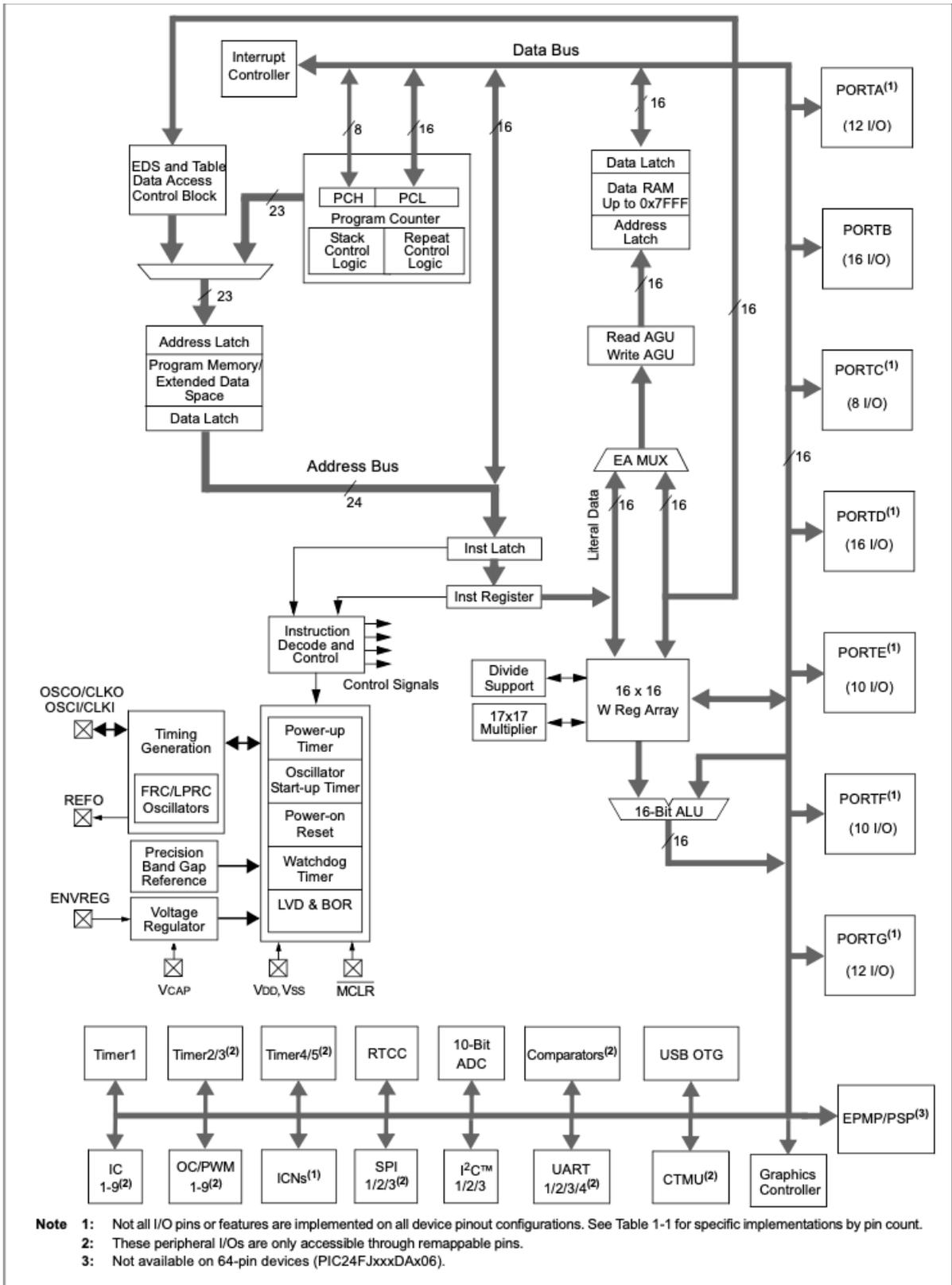


FIGURA 52: DIAGRAMA DE BLOQUES GENERAL DE LA MCU.

Pin	Function	Pin	Function
1	GCLK/CN82/RG15	41	AN12/PMA11/CTEDG2/CN30/RB12
2	Vdd	42	AN13/PMA10/CTEDG1/CN31/RB13
3	PMD5/CN63/RE5	43	AN14/CTPLS/RP14/PMA1/CN32/RB14
4	SCL3/PMD6/CN64/RE6	44	AN15/REFO/RP29/PMA0/CN12/RB15
5	SDA3/PMD7/CN65/RE7	45	Vss
6	RPI38/GD0/CN45/RC1	46	Vdd
7	RPI39/GD8/CN46/RC2	47	RPI43/GD14/CN20/RD14
8	RPI40/GD1/CN47/RC3	48	RP5/GD15/CN21/RD15
9	AN16/RPI41/PMCS2/PMA22 ⁽²⁾ /CN48/RC4	49	RP10/PMA9/CN17/RF4
10	AN17/C1IND/RP21/PMA5/PMA18 ⁽²⁾ /CN8/RG6	50	RP17/PMA8/CN18/RF5
11	AN18/C1INC/RP26/PMA4/PMA20 ⁽²⁾ /CN9/RG7	51	RP16/USBID/CN71/RF3
12	AN19/C2IND/RP19/PMA3/PMA21 ⁽²⁾ /CN10/RG8	52	RP30/GD3/CN70/RF2
13	MCLR	53	RP15/GD9/CN74/RF8
14	AN20/C2INC/RP27/PMA2/CN11/RG9	54	Vbus/CN73/RF7
15	Vss	55	Vusb
16	Vdd	56	D-/CN84/RG3
17	TMS/CN33/RA0	57	D+/CN83/RG2
18	RPI33/PMCS1/CN66/RE8	58	SCL2/CN35/RA2
19	AN21/RPI34/PMA19/CN67/RE9	59	SDA2/PMA20/PMA4 ⁽²⁾ /CN36/RA3
20	PGEC3/AN5/C1INA/VBUSEN/RP18/CN7/RB5	60	TDI/PMA21/PMA3 ⁽²⁾ /CN37/RA4
21	PGED3/AN4/C1INB/USBOEN/RP28/GD4/CN6/RB4	61	TDO/CN38/RA5
22	AN3/C2INA/GD5/VPIO/CN5/RB3	62	Vdd
23	AN2/C2INB/VMIO/RP13/GD6/CN4/RB2	63	OSCI/CLKI/CN23/RC12
24	PGEC1/AN1/VREF ⁽¹⁾ /RP1/CN3/RB1	64	OSCO/CLKO/CN22/RC15
25	PGED1/AN0/VREF ⁽¹⁾ /RP0/CN2/RB0	65	Vss
26	PGEC2/AN6/RP6/CN24/RB6	66	SCL1/RPI36/PMA22/PMCS2 ⁽²⁾ /CN43/RA14
27	PGED2/AN7/RP7/RCV/GPWR/CN25/RB7	67	SDA1/RPI35/PMBE1/CN44/RA15
28	VREF-/PMA7/CN41/RA9	68	DMLN/RTCC/RP2/CN53/RD8
29	VREF+/PMA6/CN42/RA10	69	DPLN/RP4/GD10/PMACK2/CN54/RD9
30	AVDD	70	RP3/PMA15/PMCS2 ⁽³⁾ /CN55/RD10
31	AVSS	71	RP12/PMA14/PMCS1 ⁽³⁾ /CN56/RD11
32	AN8/RP8/GD12/CN26/RB8	72	DMH/RP11/INT0/CN49/RD0
33	AN9/RP9/GD13/CN27/RB9	73	SOSCI/C3IND/CN1/RC13
34	AN10/CVREF/PMA13/CN28/RB10	74	SOSCO/SCLKI/T1CK/C3INC/RPI37/CN0/RC14
35	AN11/PMA12/CN29/RB11	75	Vss
36	Vss	76	VCPCON/RP24/GD7/VBUSCHG/CN50/RD1
37	Vdd	77	DPH/RP23/GD11/PMACK1/CN51/RD2
38	TCK/CN34/RA1	78	RP22/PMBE0/CN52/RD3
39	RP31/GD2/CN76/RF13	79	RPI42/PMD12/CN57/RD12
40	RPI32/PMA18/PMA5 ⁽²⁾ /CN75/RF12	80	PMD13/CN19/RD13

Pin	Function	Pin	Function
81	RP25/PMWR/CN13/RD4	91	AN23/GEN/CN39/RA6
82	RP20/PMRD/CN14/RD5	92	AN22/PMA17/CN40/RA7
83	C3INB/PMD14/CN15/RD6	93	PMD0/CN58/RE0
84	C3INA/SESEND/PMD15/CN16/RD7	94	PMD1/CN59/RE1
85	Vcap	95	PMA16/CN81/RG14
86	ENVREG	96	Vsync/CN79/RG12
87	VBUSST/VCMPST1/VBUSVLD/PMD11/CN68/RF0	97	HSYNC/CN80/RG13
88	VCMPST2/SESSVLD/PMD10/CN69/RF1	98	PMD2/CN60/RE2
89	PMD9/CN78/RG1	99	PMD3/CN61/RE3
90	PMD8/CN77/RG0	100	PMD4/CN62/RE4

Tabla 20: Pin- Planner del PIC24FJ256DA210.

3.7. COMPONENTES ALLEGRO MICROSYSTEMS:

Los componentes escogidos para este proyecto del fabricante Allegro Microsystems son los siguientes:

- Sensor Hall de corriente: ACS78ILLRTR.
- Amplificador de aislamiento integrado en el circuito de potencia: ACPL-C87A-500E.

Este fabricante también cumple con todas las condiciones técnicas específicas necesarias para poder integrar los componentes en las placas y garantizar un buen funcionamiento de los mismos. Debido a estas garantías de funcionamiento, el fabricante ha obtenido a lo largo de los años varios certificados de calidad y de medio ambiente cumpliendo con las normas UNE, ISO, RoHS y REACH.



FIGURA 53: COMPONENTES ALLEGRO MICROSYSTEMS.

3.8. COMPONENTES ON SEMICONDUCTOR:

Los componentes escogidos para este proyecto del fabricante ON Semiconductor son los siguientes:

- MOSFET integrado en el circuito de la barra de led's: BS170/TO.
- Diodo parásito ultra rápido integrado en el circuito de potencia: MUR460.

Estos dos cumplen también con los estándares de calidad ISO y con los de media ambiente RoHS y REACH. Garantizando dicho fabricante un nivel de calidad alto y un correcto funcionamiento dentro de las especificaciones técnicas del producto.



FIGURA 54: COMPONENTES ON SEMICONDUCTOR

3.9. COMPONENTES RECOM POWER:

Los componentes escogidos para este proyecto del fabricante ON Semiconductor son los siguientes:

- Convertidor DC/DC no aislado de salida a 5V: RPMB5.0-2.0.
- Convertidor aislado integrado en la placa de potencia: RJZ-3.315S.

Este fabricante también cumple con todas las condiciones técnicas específicas necesarias para poder integrar los componentes en las placas y garantizar un buen funcionamiento de los mismos. Debido a estas garantías de funcionamiento, el fabricante ha obtenido a lo largo de los años varios certificados de calidad y de medio ambiente cumpliendo con las normas UNE, ISO, RoHS y REACH.



FIGURA 55: ALIMENTACIONES RECOM POWER.

3.10. SOFTWARE:

Los programas que se han utilizado para llevar a cabo este proyecto son los siguientes:

- Compatible con el sistema operativo de Apple (macOS Catalina):
 - Google Chrome.
 - Pages.
 - TeamViewer.
 - Atom.
 - Matlab R2015b.

- Compatible con el sistema operativo de Microsoft (Windows 10):
 - Microsoft Word 2016.
 - OrCAD Capture.
 - OrCAD LTSpice.
 - PowerPoint 2016.



ANEXO 1: DOCUMENTACIÓN TÉCNICA



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Photocouplers Photorelay

TLP240GA, TLP240GAF

1. Applications

- Mechanical relay replacements
- Security Systems
- Measuring Instruments
- Factory Automation (FA)
- Amusement Equipment
- Smart Meters
- Electricity Meters

2. General

The TLP240GA and TLP240GAF photorelay consist of a photo MOSFET optically coupled to an infrared LED. They are housed in a 4-pin DIP package. They provide an isolation voltage of 5000 Vrms, making them suitable for applications that require reinforced insulation.

3. Features

- (1) Halogen-free
For details, see "Devices in Halogen-Free Resin Packages" at the end of this datasheet.
- (2) Normally opened (1-Form-A)
- (3) OFF-state output terminal voltage: 400 V (min)
- (4) Trigger LED current: 3 mA (max)
- (5) ON-state current: 120 mA (max)
- (6) ON-state resistance: 28 Ω (max, $t < 1$ s), 35 Ω (max, Continuous)
- (7) Isolation voltage: 5000 Vrms (min)
- (8) Safety standards
 UL-recognized: UL 1577, File No.E67349
 cUL-recognized: CSA Component Acceptance Service No.5A File No.E67349
 UL-recognized: UL 508, File No.E499232 (**Note 1**)
 CQC-approved: GB4943.1, GB8898 Japan Factory
 VDE-approved: EN 60747-5-5, EN 62368-1 (**Note 2**)

Note 1: Please refer Absolute Maximum Ratings (UL-recognized UL 508) for UL 508 products.

Note 2: When a VDE approved type is needed, please designate the **Option (D4)**.

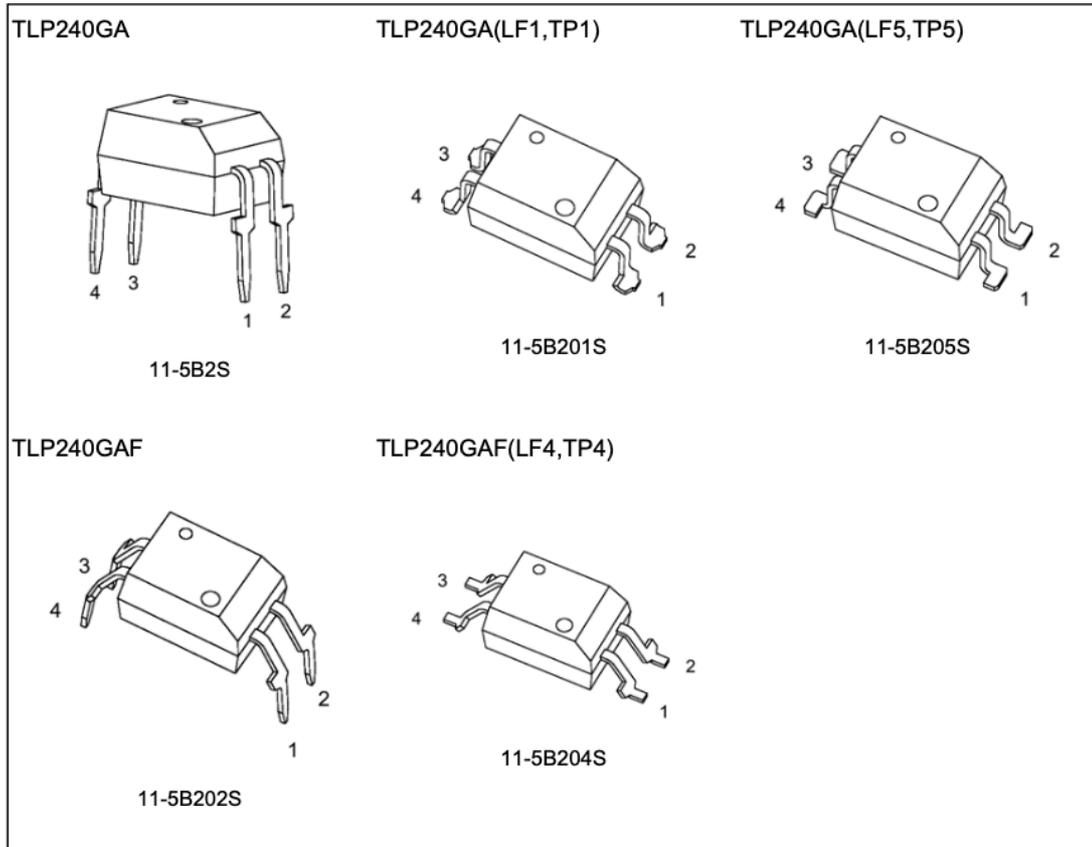
4. Mechanical Parameters

Characteristics	7.62-mm Pitch TLP240GA	10.16-mm Pitch TLP240GAF	Unit
Creepage distances	7.0 (min)	8.0 (min)	mm
Clearance distances	7.0 (min)	8.0 (min)	
Internal isolation thickness	0.4 (min)	0.4 (min)	

Start of commercial production

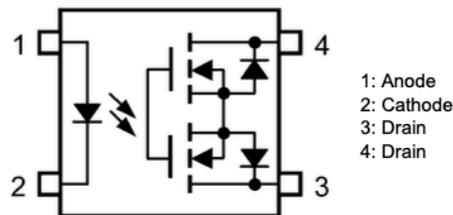
2013-02

5. Packaging (Note)

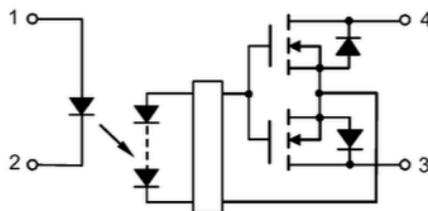


Note: Through-hole type: TLP240GA, TLP240GAF
 Lead forming option: (LF1), (LF4), (LF5)
 Taping option: (TP1), (TP4), (TP5)

6. Pin Assignment



7. Internal Circuit



8. Absolute Maximum Ratings (Note) (Unless otherwise specified, $T_a = 25\text{ }^\circ\text{C}$)

	Characteristics	Symbol	Note	Rating	Unit
LED	Input forward current	I_F		30	mA
	Input forward current derating ($T_a \geq 25\text{ }^\circ\text{C}$)	$\Delta I_F / \Delta T_a$		-0.3	mA/ $^\circ\text{C}$
	Input forward current (pulsed) (100 μs pulse, 100 pps)	I_{FP}		1	A
	Input reverse voltage	V_R		5	V
	Input power dissipation	P_D		50	mW
	Input power dissipation derating ($T_a \geq 25\text{ }^\circ\text{C}$)	$\Delta P_D / \Delta T_a$		-0.5	mW/ $^\circ\text{C}$
	Junction temperature	T_j		125	$^\circ\text{C}$
Detector	OFF-state output terminal voltage	V_{OFF}		400	V
	ON-state current	I_{ON}		120	mA
	ON-state current derating ($T_a \geq 25\text{ }^\circ\text{C}$)	$\Delta I_{ON} / \Delta T_a$		-1.2	mA/ $^\circ\text{C}$
	ON-state current (pulsed) (t = 100 ms, Duty = 1/10)	I_{ONP}		360	mA
	Output power dissipation	P_O		500	mW
	Output power dissipation derating ($T_a \geq 25\text{ }^\circ\text{C}$)	$\Delta P_O / \Delta T_a$		-5.0	mW/ $^\circ\text{C}$
	Junction temperature	T_j		125	$^\circ\text{C}$
Common	Storage temperature	T_{stg}		-55 to 125	
	Operating temperature	T_{opr}		-40 to 85	
	Lead soldering temperature (10 s)	T_{sol}		260	
	Isolation voltage (AC, 60 s, R.H. $\leq 60\%$)	BV_S	(Note 1)	5000	Vrms

Note: Using continuously under heavy loads (e.g. the application of high temperature/current/voltage and the significant change in temperature, etc.) may cause this product to decrease in the reliability significantly even if the operating conditions (i.e. operating temperature/current/voltage, etc.) are within the absolute maximum ratings.

Please design the appropriate reliability upon reviewing the Toshiba Semiconductor Reliability Handbook ("Handling Precautions"/"Derating Concept and Methods") and individual reliability data (i.e. reliability test report and estimated failure rate, etc).

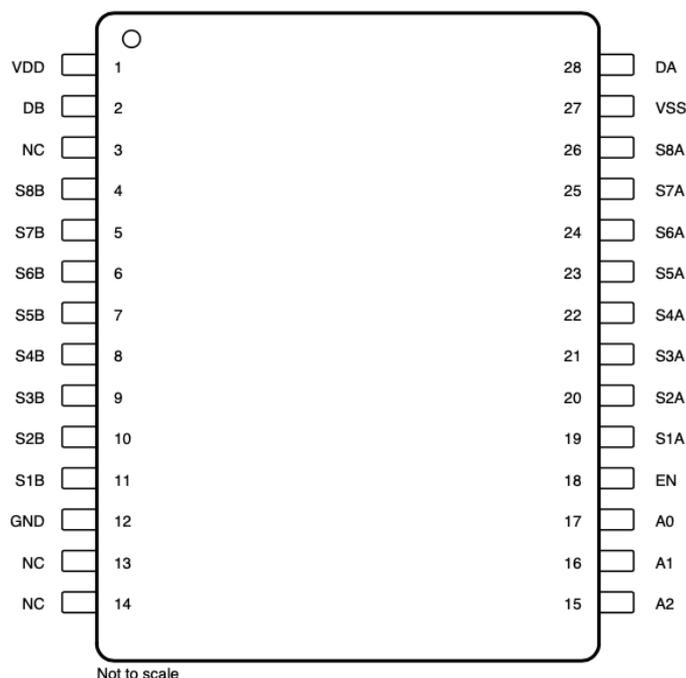
Note 1: This device is considered as a two-terminal device: Pins 1 and 2 are shorted together, and pins 3 and 4 are shorted together.

9. Recommended Operating Conditions (Note)

Characteristics	Symbol	Note	Min	Typ.	Max	Unit
Supply voltage	V_{DD}		—	—	320	V
Input forward current	I_F		5	7.5	25	mA
ON-state current	I_{ON}		—	—	120	mA
Operating temperature	T_{opr}		-20	—	65	$^\circ\text{C}$

Note: The recommended operating conditions are given as a design guide necessary to obtain the intended performance of the device. Each parameter is an independent value. When creating a system design using this device, the electrical characteristics specified in this data sheet should also be considered.

**MUX507: PW and DW Package
28-Pin TSSOP and SOIC
Top View**



Not to scale

Pin Functions: MUX507

PIN		FUNCTION	DESCRIPTION
NAME	NO.		
A0	17	Digital input	Address line 0
A1	16	Digital input	Address line 1
A2	15	Digital input	Address line 2
DA	28	Analog input or output	Drain pin A. Can be an input or output.

Pin Functions: MUX507 (continued)

PIN		FUNCTION	DESCRIPTION
NAME	NO.		
DB	2	Analog input or output	Drain pin B. Can be an input or output.
EN	18	Digital input	Active high digital input. When this pin is low, all switches are turned off. When this pin is high, the A[2:0] logic inputs determine which pair of switches is turned on.
GND	12	Power supply	Ground (0 V) reference
NC	3, 13, 14	No connect	Do not connect
S1A	19	Analog input or output	Source pin 1A. Can be an input or output.
S2A	20	Analog input or output	Source pin 2A. Can be an input or output.
S3A	21	Analog input or output	Source pin 3A. Can be an input or output.
S4A	22	Analog input or output	Source pin 4A. Can be an input or output.
S5A	23	Analog input or output	Source pin 5A. Can be an input or output.
S6A	24	Analog input or output	Source pin 6A. Can be an input or output.
S7A	25	Analog input or output	Source pin 7A. Can be an input or output.
S8A	26	Analog input or output	Source pin 8A. Can be an input or output.
S1B	11	Analog input or output	Source pin 1B. Can be an input or output.
S2B	10	Analog input or output	Source pin 2B. Can be an input or output.
S3B	9	Analog input or output	Source pin 3B. Can be an input or output.
S4B	8	Analog input or output	Source pin 4B. Can be an input or output.
S5B	7	Analog input or output	Source pin 5B. Can be an input or output.
S6B	6	Analog input or output	Source pin 6B. Can be an input or output.
S7B	5	Analog input or output	Source pin 7B. Can be an input or output.
S8B	4	Analog input or output	Source pin 8B. Can be an input or output.
VDD	1	Power supply	Positive power supply. This pin is the most positive power supply potential. For reliable operation, connect a decoupling capacitor ranging from 0.1 μ F to 10 μ F between VDD and GND.
VSS	27	Power supply	Negative power supply. This pin is the most negative power supply potential. In single-supply applications, this pin can be connected to ground. For reliable operation, connect a decoupling capacitor ranging from 0.1 μ F to 10 μ F between VSS and GND.

MUX506, MUX507

SBAS803A – NOVEMBER 2016 – REVISED NOVEMBER 2017

www.ti.com

6 Specifications

6.1 Absolute Maximum Ratings

 over operating free-air temperature range (unless otherwise noted)⁽¹⁾

		MIN	MAX	UNIT	
Voltage	Supply	V_{DD}	-0.3	40	V
		V_{SS}	-40	0.3	
		$V_{DD} - V_{SS}$		40	
	Digital pins ⁽²⁾ : EN, A0, A1, A2, A3	$V_{SS} - 0.3$	$V_{DD} + 0.3$		
	Analog pins ⁽²⁾ : Sx, SxA, SxB, D, DA, DB	$V_{SS} - 2$	$V_{DD} + 2$		
Current ⁽³⁾		-30	30	mA	
Temperature	Operating, T_A	-55	150	°C	
	Junction, T_J		150		
	Storage, T_{stg}	-65	150		

(1) Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, which do not imply functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions*. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

(2) Voltage limits are valid if current is limited to ± 30 mA.

(3) Only one pin at a time.

6.2 ESD Ratings

		VALUE	UNIT	
$V_{(ESD)}$	Electrostatic discharge	Human-body model (HBM), per ANSI/ESDA/JEDEC JS-001 ⁽¹⁾	2000	V
		Charged-device model (CDM), per JEDEC specification JESD22-C101 ⁽²⁾	500	

(1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.

(2) JEDEC document JEP157 states that 250-V CDM allows safe manufacturing with a standard ESD control process.

6.3 Recommended Operating Conditions

		MIN	NOM	MAX	UNIT
V_{DD} ⁽¹⁾	Positive power-supply voltage	Dual supply	5	18	V
		Single supply	10	36	
V_{SS} ⁽²⁾	Negative power-supply voltage (dual supply)	-5		-18	V
$V_{DD} - V_{SS}$	Supply voltage	10		36	V
V_S	Source pins voltage ⁽³⁾	V_{SS}		V_{DD}	V
V_D	Drain pins voltage	V_{SS}		V_{DD}	V
V_{EN}	Enable pin voltage	V_{SS}		V_{DD}	V
V_A	Address pins voltage	V_{SS}		V_{DD}	V
I_{CH}	Channel current ($T_A = 25^\circ\text{C}$)	-25		25	mA
T_A	Operating temperature	-40		125	°C

(1) When $V_{SS} = 0$ V, V_{DD} can range from 10 V to 36 V.

(2) V_{DD} and V_{SS} can be any value as long as $10\text{ V} \leq (V_{DD} - V_{SS}) \leq 36\text{ V}$.

(3) V_S is the voltage on all the S pins.



Low Power, Wide Supply Range, Low Cost Unity-Gain Difference Amplifiers

Data Sheet

AD8276/AD8277

FEATURES

- Wide input range beyond supplies
- Rugged input overvoltage protection
- Low supply current: 200 μ A maximum per channel
- Low power dissipation: 0.54 mW at $V_S = 2.7$ V
- Bandwidth: 550 kHz
- CMRR: 86 dB minimum, dc to 10 kHz
- System offset voltage: ± 2 μ V/ $^{\circ}$ C maximum (B Grade)
- Low gain drift: 1 ppm/ $^{\circ}$ C maximum (B Grade)
- Enhanced slew rate: 1.1 V/ μ s
- Wide power supply range
 - Single supply: 2.0 V to 36 V
 - Dual supplies: ± 2 V to ± 18 V

APPLICATIONS

- Voltage measurement and monitoring
- Current measurement and monitoring
- Differential output instrumentation amplifier
- Portable, battery-powered equipment
- Test and measurement

GENERAL DESCRIPTION

The AD8276/AD8277 are general-purpose, unity-gain difference amplifiers intended for precision signal conditioning in power critical applications that require both high performance and low power. They provide exceptional 86 dB common-mode rejection ratio (CMRR) and high bandwidth while amplifying signals well beyond the supply rails. The on-chip resistors are laser trimmed for gain drift of 1 ppm/ $^{\circ}$ C and high CMRR. The AD8276/AD8277 also have extremely low gain drift vs. temperature.

The common-mode range of the amplifiers extends to almost double the supply voltage, making these amplifiers ideal for single-supply applications that require a high common-mode voltage range. The internal resistors and electrostatic discharge (ESD) circuitry at the inputs also provide overvoltage protection to the op amps.

The AD8276/AD8277 are unity-gain stable. Although they are optimized for use as difference amplifiers, they can also be connected in high precision, single-ended configurations with $G = -1$, $+1$, or $+2$. The AD8276/AD8277 provide an integrated precision solution that has smaller size, lower cost, and improved performance than a discrete alternative.

The AD8276/AD8277 operate on single supplies (2.0 V to 36 V) or dual supplies (± 2 V to ± 18 V). The maximum quiescent supply current is 200 μ A per channel, which is ideal for battery-operated and portable systems.

Rev. D

[Document Feedback](#)

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

FUNCTIONAL BLOCK DIAGRAMS

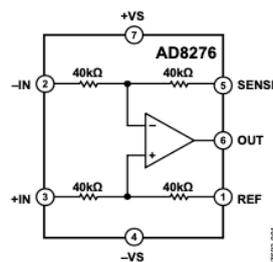


Figure 1. AD8276

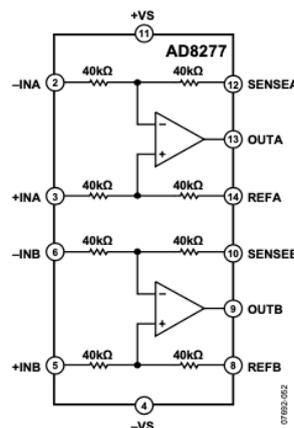


Figure 2. AD8277

The AD8276 is available in the space-saving 8-lead mini small outline package (MSOP) and the standard small outline (SOIC) package, as well as in die form, and the AD8277 is offered in a 14-lead SOIC package. Both are specified for performance over the industrial temperature range of -40° C to $+85^{\circ}$ C and are fully RoHS compliant.

Table 1. Difference Amplifiers by Category

Low Distortion	High Voltage	Current Sensing ¹	Low Power
AD8270	AD628	AD8202 (U)	AD8276
AD8271	AD629	AD8203 (U)	AD8277
AD8273		AD8205 (B)	AD8278
AD8274		AD8206 (B)	
AMP03		AD8216 (B)	

¹ U means unidirectional, B means bidirectional.

PIN CONFIGURATIONS AND FUNCTION DESCRIPTIONS



Figure 4. AD8276 8-Lead MSOP Pin Configuration



Figure 5. AD8276 8-Lead SOIC Pin Configuration

Table 6. AD8276 Pin Function Descriptions

Pin No.	Mnemonic	Description
1	REF	Reference Voltage Input.
2	-IN	Inverting Input.
3	+IN	Noninverting Input.
4	-VS	Negative Supply.
5	SENSE	Sense Terminal.
6	OUT	Output.
7	+VS	Positive Supply.
8	NC	No Connect. This pin is not internally connected.

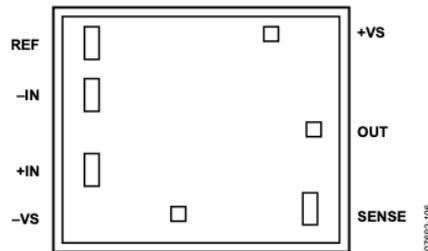


Figure 6. AD8276 7-Pad CHIP Pad Configuration

Table 7. AD8276 Pad Function Descriptions¹

Pad No.	Mnemonic	X Coordinate	Y Coordinate	Description
1	REF	-512	+412	Reference Voltage Input
2	-IN	-512	+167	Inverting Input
3	+IN	-512	-188	Noninverting Input
4	-VS	-106	-400	Negative Supply
5	SENSE	+511	-376	Sense Terminal
6	OUT	+530	0	Output
7	+VS	+329	+453	Positive Supply

¹ All dimensions are referenced from the center of the die to the center of each bond pad.

$V_S = +2.7\text{ V}$ to $\pm 5\text{ V}$, $V_{REF} = \text{midsupply}$, $T_A = 25^\circ\text{C}$, $R_L = 10\text{ k}\Omega$ connected to midsupply, $G = 1$ difference amplifier configuration, unless otherwise noted.

Table 3.

Parameter	Test Conditions/Comments	Grade B			Grade A			Unit
		Min	Typ	Max	Min	Typ	Max	
INPUT CHARACTERISTICS								
System Offset ¹			100	200		100	500	μV
vs. Temperature	$T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$			200			500	μV
Average Temperature Coefficient	$T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$		0.5	2		2	5	$\mu\text{V}/^\circ\text{C}$
vs. Power Supply	$V_S = \pm 5\text{ V}$ to $\pm 18\text{ V}$			5			10	$\mu\text{V}/\text{V}$
Common-Mode Rejection Ratio (RTI)	$V_S = 2.7\text{ V}$, $V_{CM} = 0\text{ V}$ to 2.4 V , $R_S = 0\ \Omega$	86			80			dB
	$V_S = \pm 5\text{ V}$, $V_{CM} = -10\text{ V}$ to $+7\text{ V}$, $R_S = 0\ \Omega$	86			80			dB
Input Voltage Range ²		$-2(V_S + 0.1)$		$+2(V_S - 1.5)$	$-2(V_S + 0.1)$		$+2(V_S - 1.5)$	V
Impedance ³								
Differential			80			80		k Ω
Common Mode			40			40		k Ω
DYNAMIC PERFORMANCE								
Bandwidth			450			450		kHz
Slew Rate			1.0			1.0		V/ μs
Settling Time to 0.01%	8 V step on output, $C_L = 100\text{ pF}$, $V_S = 10\text{ V}$		5			5		μs
Channel Separation	$f = 1\text{ kHz}$		130			130		dB
GAIN								
Gain Error			0.005	0.02		0.01	0.05	%
Gain Drift	$T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$			1			5	ppm/ $^\circ\text{C}$
OUTPUT CHARACTERISTICS								
Output Swing ⁴	$R_L = 10\text{ k}\Omega$, $T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$	$-V_S + 0.1$		$+V_S - 0.15$	$-V_S + 0.1$		$+V_S - 0.15$	V
Short-Circuit Current Limit			± 10			± 10		mA
Capacitive Load Drive			200			200		pF
NOISE⁵								
Output Voltage Noise	$f = 0.1\text{ Hz}$ to 10 Hz		2			2		$\mu\text{V p-p}$
	$f = 1\text{ kHz}$		65			65		nV/ $\sqrt{\text{Hz}}$
POWER SUPPLY								
Supply Current ^{6,7}	$T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$			200			200	μA
Operating Voltage Range		2.0		36	2.0		36	V
TEMPERATURE RANGE								
Operating Range		-40		+125	-40		+125	$^\circ\text{C}$

¹ Includes input bias and offset current errors, RTO.

² The input voltage range may also be limited by absolute maximum input voltage or by the output swing. See the Input Voltage Range section in the Theory of Operation section for details.

³ Internal resistors are trimmed to be ratio matched and have $\pm 20\%$ absolute accuracy.

⁴ Output voltage swing varies with supply voltage and temperature. See Figure 19 through Figure 22 for details.

⁵ Includes amplifier voltage and current noise, as well as noise from internal resistors.

⁶ Supply current varies with supply voltage and temperature. See Figure 24 and Figure 25 for details.

⁷ Power dissipation (P_{DISS}) at quiescent condition is computed by multiplying the supply voltage and quiescent supply current (I_Q). $P_{DISS} = V_S \times I_Q$, $P_{DISS} = 2.7\text{ V} \times 200\ \mu\text{A}$, $P_{DISS} = 0.54\text{ mW}$.

SPECIFICATIONS

ELECTRICAL CHARACTERISTICS—5 V, 105°C OPERATION¹

4.5 V ≤ V_{DD1} ≤ 5.5 V, 4.5 V ≤ V_{DD2} ≤ 5.5 V; all minimum/maximum specifications apply over the entire recommended operation range, unless otherwise noted; all typical specifications are at T_A = 25°C, V_{DD1} = V_{DD2} = 5 V. These specifications do not apply to [ADuM1400W](#), [ADuM1401W](#), and [ADuM1402W](#) automotive grade versions.

Table 1.

Parameter	Symbol	Min	Typ	Max	Unit	Test Conditions
DC SPECIFICATIONS						
Input Supply Current per Channel, Quiescent	I _{DD1(Q)}		0.50	0.53	mA	
Output Supply Current per Channel, Quiescent	I _{DDO(Q)}		0.19	0.21	mA	
ADuM1400 Total Supply Current, Four Channels ²						
DC to 2 Mbps						
V _{DD1} Supply Current	I _{DD1(Q)}		2.2	2.8	mA	DC to 1 MHz logic signal freq.
V _{DD2} Supply Current	I _{DD2(Q)}		0.9	1.4	mA	DC to 1 MHz logic signal freq.
10 Mbps (BRW and CRW Grades Only)						
V _{DD1} Supply Current	I _{DD1(10)}		8.6	10.6	mA	5 MHz logic signal freq.
V _{DD2} Supply Current	I _{DD2(10)}		2.6	3.5	mA	5 MHz logic signal freq.
90 Mbps (CRW Grade Only)						
V _{DD1} Supply Current	I _{DD1(90)}		70	100	mA	45 MHz logic signal freq.
V _{DD2} Supply Current	I _{DD2(90)}		18	25	mA	45 MHz logic signal freq.
ADuM1401 Total Supply Current, Four Channels ²						
DC to 2 Mbps						
V _{DD1} Supply Current	I _{DD1(Q)}		1.8	2.4	mA	DC to 1 MHz logic signal freq.
V _{DD2} Supply Current	I _{DD2(Q)}		1.2	1.8	mA	DC to 1 MHz logic signal freq.
10 Mbps (BRW and CRW Grades Only)						
V _{DD1} Supply Current	I _{DD1(10)}		7.1	9.0	mA	5 MHz logic signal freq.
V _{DD2} Supply Current	I _{DD2(10)}		4.1	5.0	mA	5 MHz logic signal freq.
90 Mbps (CRW Grade Only)						
V _{DD1} Supply Current	I _{DD1(90)}		57	82	mA	45 MHz logic signal freq.
V _{DD2} Supply Current	I _{DD2(90)}		31	43	mA	45 MHz logic signal freq.
ADuM1402 Total Supply Current, Four Channels ²						
DC to 2 Mbps						
V _{DD1} or V _{DD2} Supply Current	I _{DD1(Q)} , I _{DD2(Q)}		1.5	2.1	mA	DC to 1 MHz logic signal freq.
10 Mbps (BRW and CRW Grades Only)						
V _{DD1} or V _{DD2} Supply Current	I _{DD1(10)} , I _{DD2(10)}		5.6	7.0	mA	5 MHz logic signal freq.
90 Mbps (CRW Grade Only)						
V _{DD1} or V _{DD2} Supply Current	I _{DD1(90)} , I _{DD2(90)}		44	62	mA	45 MHz logic signal freq.
For All Models						
Input Currents	I _{IA} , I _{IB} , I _{IC} , I _{ID} , I _{E1} , I _{E2}	-10	+0.01	+10	μA	0 V ≤ V _{IA} , V _{IB} , V _{IC} , V _{ID} ≤ V _{DD1} or V _{DD2} , 0 V ≤ V _{E1} , V _{E2} ≤ V _{DD1} or V _{DD2}
Logic High Input Threshold	V _{IH} , V _{EH}	2.0			V	
Logic Low Input Threshold	V _{IL} , V _{EL}			0.8	V	
Logic High Output Voltages	V _{OAH} , V _{OBH} , V _{OCH} , V _{ODH}	(V _{DD1} or V _{DD2}) - 0.1	5.0		V	I _{Ox} = -20 μA, V _{Ix} = V _{IxH}
		(V _{DD1} or V _{DD2}) - 0.4	4.8		V	I _{Ox} = -3.2 mA, V _{Ix} = V _{IxH}
Logic Low Output Voltages	V _{OAL} , V _{OBL} , V _{OCL} , V _{ODL}		0.0	0.1	V	I _{Ox} = 20 μA, V _{Ix} = V _{IxL}
			0.04	0.1	V	I _{Ox} = 400 μA, V _{Ix} = V _{IxL}
			0.2	0.4	V	I _{Ox} = 3.2 mA, V _{Ix} = V _{IxL}
SWITCHING SPECIFICATIONS						
ADuM1400ARW/ADuM1401ARW/ADuM1402ARW						
Minimum Pulse Width ³	PW			1000	ns	C _L = 15 pF, CMOS signal levels
Maximum Data Rate ⁴		1			Mbps	C _L = 15 pF, CMOS signal levels
Propagation Delay ⁵	t _{PHL} , t _{PLH}	50	65	100	ns	C _L = 15 pF, CMOS signal levels

FEATURES

- Enhanced system-level ESD performance per IEC 61000-4-x
- High temperature operation: 125°C
- Default low output
- Narrow body, RoHS-compliant, 8-lead SOIC
- Low power operation
 - 5 V operation
 - 1.7 mA per channel maximum at 0 Mbps to 2 Mbps
 - 3.8 mA per channel maximum at 10 Mbps
 - 7.5 mA per channel maximum at 25 Mbps
 - 3.3 V operation
 - 1.5 mA per channel maximum at 0 Mbps to 2 Mbps
 - 2.5 mA per channel maximum at 10 Mbps
 - 4.7 mA per channel maximum at 25 Mbps
- 3.3 V/5 V level translation
- High data rate: dc to 25 Mbps (NRZ)
- Precise timing characteristics
 - 3 ns maximum pulse width distortion at 5 V operation
 - 3 ns maximum channel-to-channel matching
- High common-mode transient immunity: >25 kV/μs
- Safety and regulatory approvals
 - UL recognition: 2500 V rms for 1 minute per UL 1577
 - CSA Component Acceptance Notice #5A
 - VDE certificate of conformity
 - DIN V VDE V 0884-10 (VDE V 0884-10):2006-12
 - V_{IORM} = 560 V peak
- Qualified for automotive applications

APPLICATIONS

- Size-critical multichannel isolation
- SPI interface/data converter isolation
- RS-232/RS-422/RS-485 transceiver isolation
- Digital field bus isolation
- Gate drive interfaces
- Hybrid electric vehicles, battery monitor, and motor drive

GENERAL DESCRIPTION

The ADuM3210/ADuM3211¹ are dual-channel digital isolators based on Analog Devices, Inc., *iCoupler*® technology. Combining high speed CMOS and monolithic transformer technology, this isolation component provides outstanding performance characteristics superior to alternatives such as optocoupler devices.

By avoiding the use of LEDs and photodiodes, *iCoupler* devices remove the design difficulties commonly associated with optocouplers. The typical optocoupler concerns regarding uncertain current transfer ratios, nonlinear transfer functions, and temperature and lifetime effects are eliminated with the simple *iCoupler* digital interfaces and stable performance characteristics. The need for external drivers and other discrete components is eliminated with these *iCoupler* products. Furthermore, *iCoupler* devices consume one-tenth to one-sixth the power of optocouplers at comparable signal data rates.

The ADuM3210/ADuM3211 isolators provide two independent isolation channels in two channel configurations with data rates up to 25 Mbps (see the Ordering Guide). They operate with 3.3 V or 5 V supply voltages on either side, providing compatibility with lower voltage systems, as well as enabling voltage translation functionality across the isolation barrier. The ADuM3210/ADuM3211 isolators have a default output low characteristic in comparison to the ADuM3200/ADuM3201 models, which have a default output high characteristic. The ADuM3210W and ADuM3211W models are automotive grade versions qualified for 125°C operation.

In comparison to the ADuM1200/ADuM1201 isolator, the ADuM3210/ADuM3211 isolators contain various circuit and layout changes providing increased capability relative to system-level IEC 61000-4-x testing (ESD, burst, and surge). The precise capability in these tests for either the ADuM1200/ADuM1201 or ADuM3210/ADuM3211 products is strongly determined by the design and layout of the user's board or module. For more information, see the AN-793 Application Note, *ESD/Latch-Up Considerations with iCoupler Isolation Products*.

FUNCTIONAL BLOCK DIAGRAMS

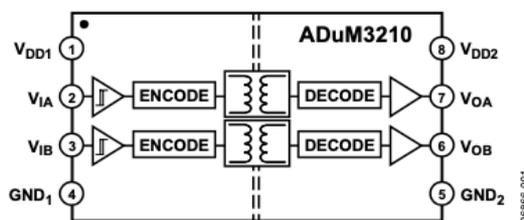


Figure 1. ADuM3210

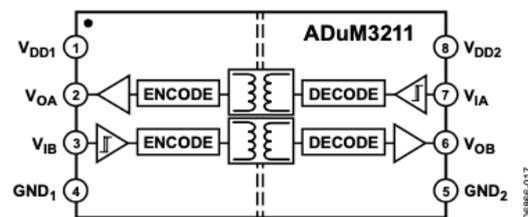


Figure 2. ADuM3211

¹ Protected by U.S. Patents 5,952,849; 6,873,065; and 7,075,239.

Rev. K Document Feedback
 Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
 Tel: 781.329.4700 ©2008–2015 Analog Devices, Inc. All rights reserved.
 Technical Support www.analog.com

ELECTRICAL CHARACTERISTICS—3.3 V, 105°C OPERATION

All typical specifications are at $T_A = 25^\circ\text{C}$, $V_{DD1} = V_{DD2} = 3.3\text{ V}$. Minimum/maximum specifications apply over the entire recommended operation range: $3.0\text{ V} \leq V_{DD1} \leq 3.6\text{ V}$, $3.0\text{ V} \leq V_{DD2} \leq 3.6\text{ V}$, and $-40^\circ\text{C} \leq T_A \leq +105^\circ\text{C}$, unless otherwise noted. Switching specifications are tested with $C_L = 15\text{ pF}$ and CMOS signal levels, unless otherwise noted.

Table 4.

Parameter	Symbol	A Grade			B Grade			Unit	Test Conditions/ Comments
		Min	Typ	Max	Min	Typ	Max		
SWITCHING SPECIFICATIONS									
Data Rate				1			10	Mbps	Within PWD limit
Propagation Delay	t_{PHL}, t_{PLH}	20		60	20		60	ns	50% input to 50% output
Pulse Width Distortion	PWD								
ADuM3210				5			3	ns	$ t_{PLH} - t_{PHL} $
ADuM3211				6			4	ns	$ t_{PLH} - t_{PHL} $
Change vs. Temperature			6			5		ps/°C	
Pulse Width	PW								Within PWD limit
ADuM3210		1000			22			ns	
ADuM3211		1000			33			ns	
Propagation Delay Skew	t_{PSK}			29			22	ns	Between any two units
Channel Matching									
Codirectional	t_{PSKCD}			5			3	ns	
Opposing Directional	t_{PSKOD}			29			20	ns	
Output Rise/Fall Time	t_R/t_F		3.0			3.0		ns	10% to 90%

Table 5.

Parameter	Symbol	1 Mbps—A Grade, B Grade			10 Mbps—B Grade			Unit
		Min	Typ	Max	Min	Typ	Max	
SUPPLY CURRENT								
ADuM3210	I_{DD1}		0.8	1.3		2.1	3.2	mA
	I_{DD2}		0.7	1.0		1.3	1.9	mA
ADuM3211	I_{DD1}		0.7	1.3		1.8	2.6	mA
	I_{DD2}		0.8	1.6		1.9	2.5	mA

Table 6. For All Models

Parameter	Symbol	Min	Typ	Max	Unit	Test Conditions/Comments
DC SPECIFICATIONS						
Logic High Input Threshold	V_{IH}	$0.7 V_{DDx}$			V	
Logic Low Input Threshold	V_{IL}				V	
Logic High Output Voltages	V_{OH}	$V_{DDx} - 0.1$	3.0	$0.3 V_{DDx}$	V	$I_{Ox} = -20\ \mu\text{A}, V_{Ix} = V_{IxH}$
		$V_{DDx} - 0.5$	2.8		V	$I_{Ox} = -3.2\ \text{mA}, V_{Ix} = V_{IxH}$
Logic Low Output Voltages	V_{OL}	0.0			V	$I_{Ox} = 20\ \mu\text{A}, V_{Ix} = V_{IxL}$
		0.2			V	$I_{Ox} = 3.2\ \text{mA}, V_{Ix} = V_{IxL}$
Input Current per Channel	I_i	-10	+0.01	+10	μA	$0\text{ V} \leq V_{Ix} \leq V_{DDx}$
Supply Current per Channel						
Quiescent Input Supply Current	$I_{DD(IQ)}$	0.3			mA	
Quiescent Output Supply Current	$I_{DD(OQ)}$	0.3			mA	
Dynamic Input Supply Current	$I_{DD(ID)}$	0.10			mA/Mbps	
Dynamic Output Supply Current	$I_{DD(OD)}$	0.03			mA/Mbps	
AC SPECIFICATIONS						
Common-Mode Transient Immunity ¹	$ CM $	25	35		kV/ μs	$V_{Ix} = V_{DDx}, V_{CM} = 1000\text{ V}$, transient magnitude = 800 V
Refresh Rate	f_r	1.1			Mbps	

¹ $|CM|$ is the maximum common-mode voltage slew rate that can be sustained while maintaining $V_O > 0.8 V_{DD}$. The common-mode voltage slew rates apply to both rising and falling common-mode voltage edges.

REF33xx 3.9- μ A, SC70-3, SOT-23-3, and UQFN-8, 30-ppm/ $^{\circ}$ C Drift Voltage Reference

1 Features

- Microsize Packages: SC70-3, SOT-23-3, UQFN-8
- Low Supply Current: 3.9 μ A (typ)
- Extremely Low Dropout Voltage: 110 mV (typ)
- High Output Current: \pm 5 mA
- Low Temperature Drift: 30 ppm/ $^{\circ}$ C (max)
- High Initial Accuracy: \pm 0.15% (max)
- 0.1-Hz to 10-Hz Noise: 35 μ V_{PP} (REF3312)
- Voltage Options: 1.2 V, 1.8 V, 2.5 V, 3 V, 3.3 V

2 Applications

- Portable Equipment
- Tablets and Smartphones
- Hard Disk Drives
- Sensor Modules
- Data Acquisition Systems
- Medical Equipment
- Test Equipment

3 Description

The REF33xx is a low-power, precision, low-dropout voltage reference family available in tiny SC70-3 and SOT-23-3 packages, and in a 1.5-mm \times 1.5-mm UQFN-8 package. Small size and low power consumption (5- μ A max) make the REF33xx ideal for a wide variety of portable and battery-powered applications.

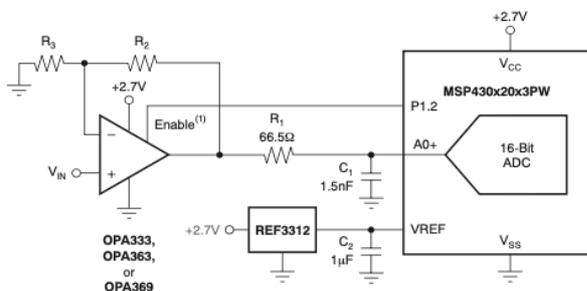
The REF33xx can be operated at a supply voltage 180 mV above the specified output voltage under normal load conditions, with the exception of the REF3312, which has a minimum supply voltage of 1.7 V. All models are specified for the wide temperature range of -40° C to $+125^{\circ}$ C.

Device Information⁽¹⁾

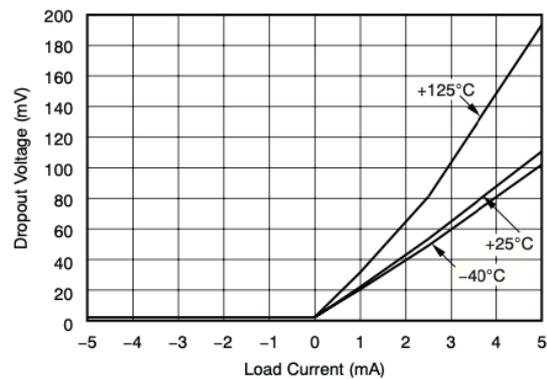
PART NUMBER	PACKAGE	BODY SIZE (NOM)
REF33xx	SOT-23 (3)	2.92 mm \times 1.30 mm
	SC70 (3)	2.00 mm \times 1.25 mm
	UQFN (8)	1.50 mm \times 1.50 mm

(1) For all available packages, see the package option addendum at the end of the datasheet.

REF3312 in a Single-Supply Signal Chain



Dropout Voltage vs Load Current

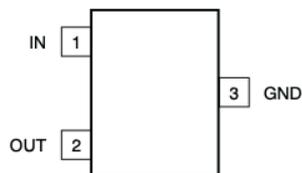


5 Device Comparison

PRODUCT	DESCRIPTION
REF3312	1.25 V
REF3318	1.8 V
REF3320	2.048 V
REF3325	2.5 V
REF3330	3.0 V
REF3333	3.3 V

6 Pin Configuration and Functions

REF3312, REF3318, REF3320, REF3325, REF3330, REF3333
 DBZ Package and DCK Package
 SOT-23-3, SC70-3
 (Top View)



REF3312, REF3318, REF3320, REF3325, REF3330, REF3333
 RSE Package
 UQFN-8
 (Top View)

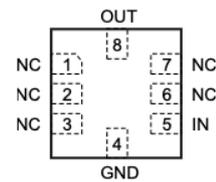


Table 1. Pin Functions

NAME	PIN		DESCRIPTION
	DBZ, DCK	RSE	
GND	3	4	Ground
IN	1	5	Input supply voltage
NC	—	1, 2, 3, 6, 7	Not connected
OUT	2	8	Output voltage

7 Specifications

7.1 Absolute Maximum Ratings

over operating free-air temperature range (unless otherwise noted).⁽¹⁾

		MIN	MAX	UNIT
Voltage	Input voltage		7.5	V
	Output voltage		5	
Current	Output short-circuit, I_{SC} ⁽²⁾		180	mA
Temperature	Operating	-50	150	°C
	Junction, T_J		150	
	Storage, T_{stg}	-65	150	

- (1) Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, which do not imply functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions*. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
- (2) See the [Power-Supply Recommendations](#) section of this data sheet.

7.2 ESD Ratings

		VALUE	UNIT
V_{ESD}	Electrostatic discharge	Human body model (HBM), per ANSI/ESDA/JEDEC JS-001 ⁽¹⁾	±4000
		Charged device model (CDM), per JEDEC specification JESD22-C101 ⁽²⁾	±1000
		Machine model (MM)	±200

- (1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.
- (2) JEDEC document JEP157 states that 250-V CDM allows safe manufacturing with a standard ESD control process.

7.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted).

		MIN	NOM	MAX	UNIT
V_{IN}	Supply input voltage ⁽¹⁾	$V_{OUT} + 0.2$		5.5	V
I_{OUT}	Output current range	-5		5	mA

- (1) The minimum supply voltage for the REF3312 is 1.7 V.

7.4 Thermal Information

THERMAL METRIC ⁽¹⁾	REF33xx		REF3325, REF3330	UNIT	
	DCK (SC70)	DBZ (SOT-23)	RSE (UQFN)		
	3 PINS	3 PINS	8 PINS		
$R_{\theta JA}$	Junction-to-ambient thermal resistance	279.7	313.1	61.2	°C/W
$R_{\theta JC(top)}$	Junction-to-case (top) thermal resistance	136.3	144.0	32.6	°C/W
$R_{\theta JB}$	Junction-to-board thermal resistance	56.9	109.3	16.0	°C/W
Ψ_{JT}	Junction-to-top characterization parameter	11.0	18.2	1.3	°C/W
Ψ_{JB}	Junction-to-board characterization parameter	56.1	107.9	16.0	°C/W
$R_{\theta JC(bot)}$	Junction-to-case (bottom) thermal resistance	N/A	N/A	N/A	°C/W

- (1) For more information about traditional and new thermal metrics, see the [Semiconductor and IC Package Thermal Metrics](#) application report.



Single-Supply, 10MHz, Rail-to-Rail Output, Low-Noise, JFET Amplifier

Check for Samples: [OPA141](#), [OPA2141](#), [OPA4141](#)

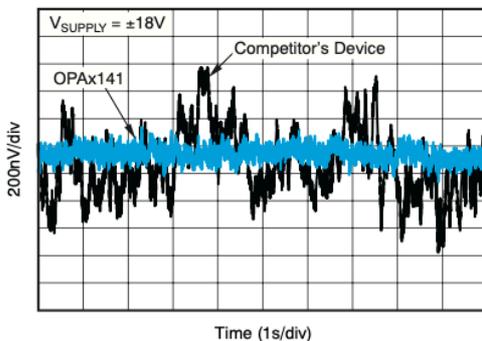
FEATURES

- **Low Supply Current:** 2.3mA max
- **Low Offset Drift:** 10 μ V/ $^{\circ}$ C max
- **Low Input Bias Current:** 20pA max
- **Very Low 1/f Noise:** 250nV_{PP}
- **Low Noise:** 6.5nV/ $\sqrt{\text{Hz}}$
- **Wide Bandwidth:** 10MHz
- **Slew Rate:** 20V/ μ s
- **Input Voltage Range Includes V₋**
- **Rail-to-Rail Output**
- **Single-Supply Operation:** 4.5V to 36V
- **Dual-Supply Operation:** \pm 2.25V to \pm 18V
- **No Phase Reversal**
- **MSOP-8, TSSOP Packages**

APPLICATIONS

- **Battery-Powered Instruments**
- **Industrial Controls**
- **Medical Instrumentation**
- **Photodiode Amplifiers**
- **Active Filters**
- **Data Acquisition Systems**
- **Portable Audio**
- **Automatic Test Systems**

0.1Hz to 10Hz NOISE



DESCRIPTION

The OPA141, OPA2141, and OPA4141 amplifier family is a series of low-power JFET input amplifiers that feature good drift and low input bias current. The rail-to-rail output swing and input range that includes V₋ allow designers to take advantage of the low-noise characteristics of JFET amplifiers while also interfacing to modern, single-supply, precision analog-to-digital converters (ADCs) and digital-to-analog converters (DACs).

The OPA141 achieves 10MHz unity-gain bandwidth and 20V/ μ s slew rate while consuming only 1.8mA (typ) of quiescent current. It runs on a single 4.5 to 36V supply or dual \pm 2.25V to \pm 18V supplies.

All versions are fully specified from -40° C to $+125^{\circ}$ C for use in the most challenging environments. The OPA141 (single) and OPA2141 (dual) versions are available in both MSOP-8 and SO-8 packages; the OPA4141 (quad) is available in the SO-14 and TSSOP-14 packages.

RELATED PRODUCTS

FEATURES	PRODUCT
Precision, Low-Power, 10MHz FET Input Industrial Op Amp	OPA140⁽¹⁾
2.2nV/ $\sqrt{\text{Hz}}$, Low-Power, 36V Operational Amplifier in SOT-23 Package	OPA209⁽¹⁾
Low-Noise, High-Precision, JFET-Input Operational Amplifier	OPA827
Low-Noise, Low I _O Precision Operational Amplifier	OPA376
High-Speed, FET-Input Operational Amplifier	OPA132

1. Preview product; estimated availability in Q3 2010.



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

All trademarks are the property of their respective owners.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of the Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 2010, Texas Instruments Incorporated

ELECTRICAL CHARACTERISTICS: $V_S = +4.5V$ to $+36V$; $\pm 2.25V$ to $\pm 18V$

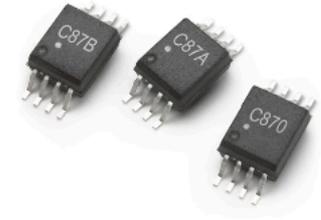
Boldface limits apply over the specified temperature range, $T_A = -40^\circ C$ to $+125^\circ C$.

At $T_A = +25^\circ C$, $R_L = 2k\Omega$ connected to midsupply, $V_{CM} = V_{OUT} =$ midsupply, unless otherwise noted.

PARAMETER	CONDITIONS	OPA141, OPA2141, OPA4141			UNIT
		MIN	TYP	MAX	
OFFSET VOLTAGE					
Offset Voltage, RTI	V_{OS}		± 1	± 3.5	mV
Over Temperature	$V_S = \pm 18V$			± 4.3	mV
Drift	dV_{OS}/dT		± 2	± 10	$\mu V/^\circ C$
vs Power Supply	PSRR	$V_S = \pm 2.25V$ to $\pm 18V$	± 0.14	± 2	$\mu V/V$
Over Temperature	$V_S = \pm 2.25V$ to $\pm 18V$			± 4	$\mu V/V$
INPUT BIAS CURRENT					
Input Bias Current	I_B		± 2	± 20	pA
Over Temperature				± 5	nA
Input Offset Current	I_{OS}		± 2	± 20	pA
Over Temperature				± 1	nA
NOISE					
Input Voltage Noise					
	$f = 0.1Hz$ to $10Hz$		250		nV _{PP}
	$f = 0.1Hz$ to $10Hz$		42		nV _{RMS}
Input Voltage Noise Density					
	e_n				
	$f = 10Hz$		12		nV/ \sqrt{Hz}
	$f = 100Hz$		6.5		nV/ \sqrt{Hz}
	$f = 1kHz$		6.5		nV/ \sqrt{Hz}
Input Current Noise Density					
	i_n				
	$f = 1kHz$		0.8		fA/ \sqrt{Hz}
INPUT VOLTAGE RANGE					
Common-Mode Voltage Range	V_{CM}		(V-) -0.1	(V+) -3.5	V
Common-Mode Rejection Ratio	CMRR	$V_S = \pm 18V, V_{CM} = (V-) -0.1V$ to $(V+) - 3.5V$	120	126	dB
Over Temperature		$V_S = \pm 18V, V_{CM} = (V-) -0.1V$ to $(V+) - 3.5V$	120		dB
INPUT IMPEDANCE					
Differential					
				$10^{13} \parallel 8$	$\Omega \parallel pF$
Common-Mode					
		$V_{CM} = (V-) -0.1V$ to $(V+) - 3.5V$		$10^{13} \parallel 6$	$\Omega \parallel pF$
OPEN-LOOP GAIN					
Open-Loop Voltage Gain	A_{OL}	$V_O = (V-) +0.35V$ to $(V+) -0.35V, R_L = 2k\Omega$	114	126	dB
Over Temperature		$V_O = (V-) +0.35V$ to $(V+) -0.35V, R_L = 2k\Omega$	108		dB
FREQUENCY RESPONSE					
Gain Bandwidth Product	BW			10	MHz
Slew Rate					
				20	V/ μs
Settling Time, 12-bit (0.024)					
				880	ns
THD+N					
		1kHz, $G = 1, V_O = 3.5V_{RMS}$		0.00005	%
Overload Recovery Time					
				600	ns

ACPL-C87B, ACPL-C87A, ACPL-C870

Precision Optically Isolated Voltage Sensor



Description

The Broadcom[®] ACPL-C87B/C87A/C870 voltage sensors are optical isolation amplifiers designed specifically for voltage sensing. Its 2V input range and high 1-G Ω input impedance, makes it well suited for isolated voltage sensing requirements in electronic power converters applications, including motor drives and renewable energy systems. In a typical voltage sensing implementation, a resistive voltage divider is used to scale the DC-link voltage to suit the input range of the voltage sensor. A differential output voltage that is proportional to the input voltage is created on the other side of the optical isolation barrier.

For general applications, the ACPL-C87A ($\pm 1\%$ gain tolerance) and the ACPL-C870 ($\pm 3\%$ gain tolerance) are recommended. For high precision requirements, the ACPL-C87B ($\pm 0.5\%$ gain tolerance) can be used. The ACPL-C87B/C87A/C870 family operates from a single 5V supply and provides excellent linearity. An active-high shutdown pin is available which reduces the I_{DD1} current to only 15A, making them suitable for battery-powered and other power-sensitive applications.

The high common-mode transient immunity (15 kV/ms) of the ACPL-C87B/C87A/C870 provides the precision and stability needed to accurately monitor DC-link voltage in high noise environments. Combined with superior optical coupling technology, the ACPL-C87B/C87A/C870 implements sigma-delta (Σ - Δ) modulation, chopper stabilized amplifiers, and differential outputs to provide unequalled isolation-mode noise rejection, low offset, high gain accuracy and stability. This performance is delivered in a compact, auto-insertable Stretched SO-8 (SSO-8) package that meets worldwide regulatory safety standards.

Features

- Advanced sigma-delta (Σ - Δ) modulation technology
- Unity gain 1 V/V, $\pm 0.5\%$ high gain accuracy (ACPL-C87B)
- 1-G Ω input impedance
- 0 to 2V nominal input range
- -35 ppm/ $^{\circ}$ C low gain frift
- 21 μ V/ $^{\circ}$ C offset voltage drift
- 0.1% non-linearity max
- Active-high shutdown pin
- 100-kHz wide bandwidth
- 3V to 5.5V wide supply range for output side
- -40° C to $+105^{\circ}$ C operating temperature range
- 15 kV/ μ s common-mode transient immunity
- Compact, auto-insertable stretched SO-8 package
- Safety and regulatory approvals:
 - IEC/EN/DIN EN 60747-5-5: 1414 V_{peak} working insulation voltage
 - UL 1577: 5000 V_{rms}/1 min double protection rating
 - CSA: Component Acceptance Notice #5

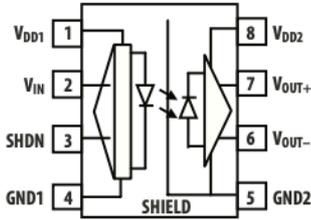
Applications

- Isolated voltage sensing in AC and servo motor drives
- Isolated DC-bus voltage sensing in solar inverters, wind turbine inverters
- Isolated sensor interfaces
- Signal isolation in data acquisition systems
- General purpose voltage isolation

CAUTION! It is advised that normal static precautions be taken in handling and assembly of this component to prevent damage and/or degradation which may be induced by ESD.

Functional Diagram

Figure 1: Functional Diagram



NOTE: A 0.1- μ F bypass capacitor must be connected between pins 1 and 4 and between pins 5 and 8.

Table 1: Pin Description

Pin No.	Symbol	Description
1	V _{DD1}	Supply voltage for input side (4.5V to 5.5V), relative to GND1
2	V _{IN}	Voltage input
3	SHDN	Shutdown pin (Active High)
4	GND1	Input side ground
5	GND2	Output side ground
6	V _{OUT-}	Negative output
7	V _{OUT+}	Positive output
8	V _{DD2}	Supply voltage for output side (3V to 5.5V), referenced to GND2

Ordering Information

ACPL-C87B/C87A/C870 is UL recognized with 5000 V_{rms}/1 minute rating per UL 1577.

Table 2: Ordering Information

Part Number	Option	Package	Surface Mount	Tape and Reel	IEC/EN/DIN EN 60747-5-5	Quantity
	(RoHS Compliant)					
ACPL-C87B	-000E	Stetched SO-8	X		X	80 per tube
ACPL-C87A	-500E		X	X	X	1000 per reel
ACPL-C870						

To order, choose a part number from the part number column and combine with the desired option from the option column to form an order entry.

Example:

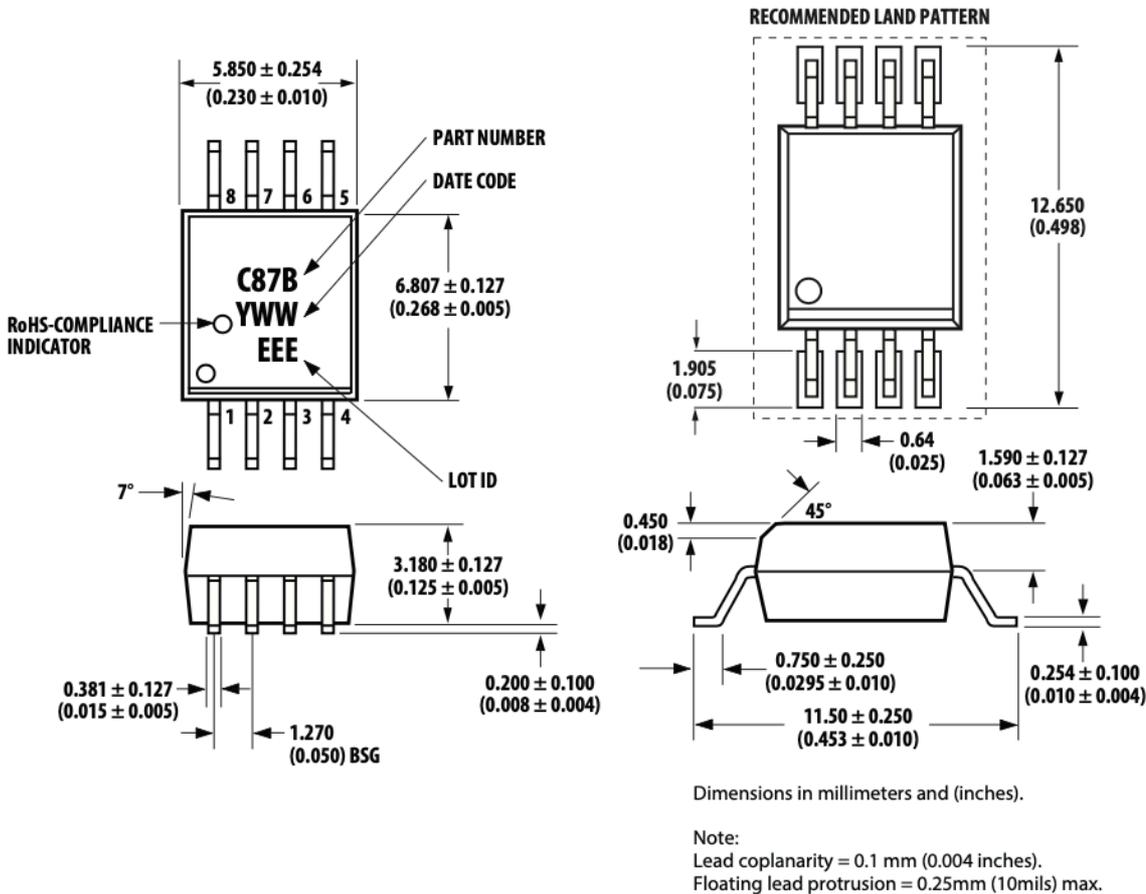
ACPL-C87A-500E to order product of Surface Mount package in Tape and Reel packaging with IEC/EN/DIN EN 60747-5-5 Safety Approval and RoHS compliance.

Contact your Broadcom sales representative or authorized distributor for information.

Package Outline Drawing

Stretched SO-8 Package (SSO-8)

Figure 2: SSO-8 Package



Recommended Pb-Free IR Profile

Recommended reflow condition as per JEDEC Standard, J-STD-020 (latest revision). Non-Halide Flux should be used.

Regulatory Information

The ACPL-C87B/C87A/C870 is approved by the following organizations.

IEC/EN/DIN EN 60747-5-5	Approval with Maximum Working Insulation Voltage $V_{IORM} = 1414 V_{peak}$.
UL	Approval under UL 1577, component recognition program up to $V_{ISO} = 5000 V_{rms}/1$ min. File 55361.
CSA	Approval under CSA Component Acceptance Notice #5, File CA 88324

Absolute Maximum Rating

Table 5: Absolute Maximum Rating

Parameter	Symbol	Min.	Max.	Units
Storage Temperature	T_S	-55	+125	°C
Ambient Operating Temperature	T_A	-40	+105	°C
Supply Voltage	V_{DD1}, V_{DD2}	-0.5	6.0	V
Steady-State Input Voltage ^{a, b}	V_{IN}	-2	$V_{DD1} + 0.5$	V
Two-Second Transient Input Voltage ^c	V_{IN}	-6	$V_{DD1} + 0.5$	V
Logic Input	V_{SD}	-0.5	$V_{DD1} + 0.5$	V
Output Voltages	V_{OUT+}, V_{OUT-}	-0.5	$V_{DD2} + 0.5$	V
Lead Solder Temperature	260° C for 10 seconds, 1.6 mm below seating plane			

a. DC voltage of up to -2V on the inputs does not cause latch-up or damage to the device.

b. Absolute maximum DC current on the inputs = 100 mA, no latch-up or device damage occurs.

c. Transient voltage of 2 seconds up to -6V on the inputs does not cause latch-up or damage to the device.

Recommended Operating Conditions

Table 6: Recommended Operating Conditions

Parameter	Symbol	Min.	Max.	Units
Ambient Operating Temperature	T_A	-40	+105	°C
V_{DD1} Supply Voltage	V_{DD1}	4.5	5.5	V
V_{DD2} Supply Voltage	V_{DD2}	3.0	5.5	V
Input Voltage Range ^a	V_{IN}	0	2.0	V
Shutdown Enable Voltage	V_{SD}	$V_{DD1} - 0.5$	V_{DD1}	V

a. 2V is the nominal input range. Full scale input range (FSR) is 2.46V.

Features

Power Module

- 36V 2A SMD Power Module
- High power density in 12.2x12.2x3.75mm case
- -40°C to +100°C with derating, convection cooled
- Efficiency up to 94%
- 6-sided shielding
- Thermally enhanced 25 pad LGA package (DOSA conform)



RPMB-2.0

2 Amp Single Output



EN55032 compliant

Description

The RPMB-2.0 series is a 2A non-isolated SMD switching regulator power module with up to 36V input voltage. Despite its compact LGA footprint and low profile (12.2x12.2x3.75mm), it offers a full set of features including adjustable output from 1V up to 24V, on/off control, sense and power good output signals. With an efficiency of up to 94% which remains nearly constant over a 5% to 100% load range, the device can operate at ambient temperatures as high as +100°C without forced air cooling. The package is complete with 6-sided shielding for optimal EMC performance and excellent heat management. The fully protected module (UVLO, SCP, OCP, OTP) can drive high capacitive loads of up to 0.2F.

Selection Guide

Part Number	Input Voltage Range [VDC]	Output Voltage [VDC]	Vout Adjust Range [VDC]	Output Current max. [A]	Efficiency typ. [%]	Max Capacitive typ. Load ⁽¹⁾ [µF]
RPMB3.3-2.0	4-36	3.3	1-9	2.0	84	200000
RPMB5.0-2.0	5.5-36	5	1-9	2.0	88	200000
RPMB12-2.0	12.8-36	12	9-24	2.0	93	10000
RPMB15-2.0	16-36	15	9-24	2.0	94	8000

Notes:

Note1: Max. Capacitive Load is tested at nominal input, nominal output, and full resistive load, below 1 second start-up

Model Numbering



Notes:

Note2: Add suffix "-CT" for tube packaging; for more details refer to "**PACKAGING INFORMATION**" without suffix, standard tape and reel packaging

Specifications (@ Ta= 25°C, nom. Vin, full load, with input cap ⁽³⁾, after warm-up unless otherwise stated)

BASIC CHARACTERISTICS					
Parameter	Condition		Min.	Typ.	Max.
Internal Input Filter			capacitor		
Input Voltage Range ⁽⁴⁾	3.3Vout		4VDC	24VDC (nominal)	36VDC
	5.0Vout		5.5VDC		
	12Vout		12.8VDC		
	15Vout		16VDC		
Absolute Maximum Input Voltage					38VDC
Input Current	nom. Vin= 24VDC	3.3Vout		0.3A	
		5.0Vout		0.5A	
		12Vout		1A	
		15Vout		1.3A	

continued on next page



www.recom-power.com/eval-ref-boards

Notes:

Note3: 4.7µF/50V/X7R input cap required

Specifications (@ Ta= 25°C, nom. Vin= 24VDC, full load, with input cap ⁽⁵⁾, after warm-up unless otherwise stated)

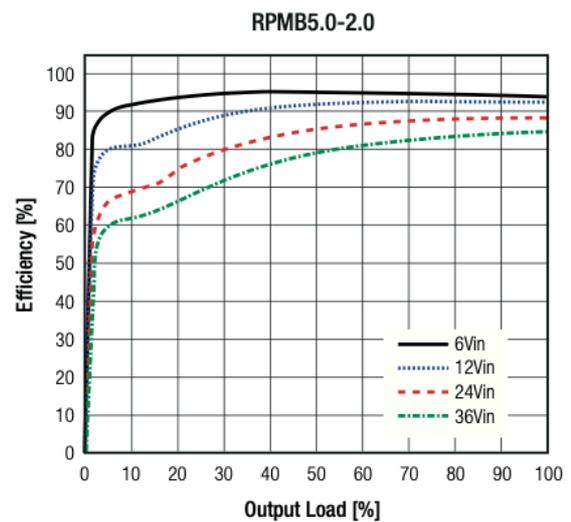
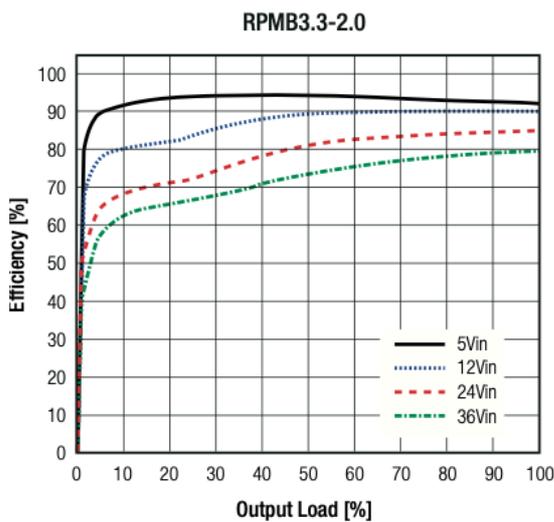
Parameter	Condition		Min.	Typ.	Max.
Quiescent Current	nom. Vin= 24VDC	3.3Vout		30µA	
		5.0Vout		36µA	
		12Vout		70µA	
		15Vout		140µA	
Internal Power Dissipation	nom. Vin= 24VDC	3.3Vout		1.3W	
		5.0Vout		1.4W	
		12Vout		1.8W	
		15Vout		1.9W	
Output Voltage Trimming	refer to "OUTPUT VOLTAGE TRIMMING"	3.3, 5.0Vout	1VDC		9VDC
		12, 15Vout	9VDC		24VDC
Minimum Load			0%		
Start-up Time	power up using CTRL function			4.8ms	
				3.8ms	
Rise-time				900µs	
ON/OFF CTRL	DC-DC ON			Open or 1.26VDC < V _{CTRL} < Vin	
	DC-DC OFF			Short to GND or -0.3VDC < V _{CTRL} < 0.3VDC	
Input Current of CTRL Pin		DC-DC OFF		25µA	
Standby Current		DC-DC OFF		35µA	
Internal Operating Frequency		for all types		1.4MHz	
Output Ripple and Noise ⁽⁵⁾	20MHz BW	3.3Vout		20mVp-p	50mVp-p
		5.0Vout		25mVp-p	60mVp-p
		12Vout		40mVp-p	90mVp-p
		15Vout		50mVp-p	100mVp-p

Notes:

Note4: Below minimum input voltage range, the module enters 98% duty cycle mode. Output voltage will not meet the output accuracy specification

Note5: Measurements are made with a 22µF MLCC across output (low ESR)

Efficiency vs. Load



continued on next page

Specifications (@ Ta= 25°C, nom. Vin= 24VDC, full load, with input cap⁽⁶⁾, after warm-up unless otherwise stated)

<p>Practical Example RPMB12-2.0</p> <p>Vout_{set} = 15VDC</p> $R_{up} = \frac{9.09 \times (15 - 1) - 100 \times (9.09 + 1)}{100 - 9.09 \times (15 - 1)}$ <p>R_{up} according to E96 ≈ 32k4Ω</p> <p>RPMB3.3-2.0</p> <p>Trim up</p> <table border="1"> <tr><td>Vout_{set} =</td><td>5</td><td>[VDC]</td></tr> <tr><td>R_{up} (E96) ≈</td><td>57k6</td><td>[Ω]</td></tr> </table> <p>Trim down</p> <table border="1"> <tr><td>Vout_{set} =</td><td>2.5</td><td>1.8</td><td>1.5</td><td>1.1</td><td>[VDC]</td></tr> <tr><td>R_{down} (E96) ≈</td><td>182k</td><td>52k3</td><td>26k7</td><td>3k48</td><td>[Ω]</td></tr> </table> <p>RPMB5.0-2.0</p> <p>Trim up</p> <table border="1"> <tr><td>Vout_{set} =</td><td>5.5</td><td>9</td><td>[VDC]</td></tr> <tr><td>R_{up} (E96) ≈</td><td>205k</td><td>23k7</td><td>[Ω]</td></tr> </table> <p>Trim down</p> <table border="1"> <tr><td>Vout_{set} =</td><td>3.3</td><td>2.5</td><td>[VDC]</td></tr> <tr><td>R_{down} (E96) ≈</td><td>133k</td><td>59k</td><td>[Ω]</td></tr> </table>	Vout _{set} =	5	[VDC]	R _{up} (E96) ≈	57k6	[Ω]	Vout _{set} =	2.5	1.8	1.5	1.1	[VDC]	R _{down} (E96) ≈	182k	52k3	26k7	3k48	[Ω]	Vout _{set} =	5.5	9	[VDC]	R _{up} (E96) ≈	205k	23k7	[Ω]	Vout _{set} =	3.3	2.5	[VDC]	R _{down} (E96) ≈	133k	59k	[Ω]	<p>Practical Example RPMB12-2.0</p> <p>Vout_{set} = 9VDC</p> $R_{down} = \frac{9.09 \times (9 - 1) \times (100 + 1) - 100}{100 - 9.09 \times (9 - 1)}$ <p>R_{down} according to E96 ≈ 267kΩ</p> <p>RPMB12-2.0</p> <p>Trim up</p> <table border="1"> <tr><td>Vout_{set} =</td><td>15</td><td>24</td><td>[VDC]</td></tr> <tr><td>R_{up} (E96) ≈</td><td>32k4</td><td>7k32</td><td>[Ω]</td></tr> </table> <p>Trim down</p> <table border="1"> <tr><td>Vout_{set} =</td><td>10</td><td>9</td><td>[VDC]</td></tr> <tr><td>R_{down} (E96) ≈</td><td>453k</td><td>267k</td><td>[Ω]</td></tr> </table> <p>RPMB15-2.0</p> <p>Trim up</p> <table border="1"> <tr><td>Vout_{set} =</td><td>20</td><td>24</td><td>[VDC]</td></tr> <tr><td>R_{up} (E96) ≈</td><td>16k9</td><td>9k09</td><td>[Ω]</td></tr> </table> <p>Trim down</p> <table border="1"> <tr><td>Vout_{set} =</td><td>12</td><td>9.99</td><td>[VDC]</td></tr> <tr><td>R_{down} (E96) ≈</td><td>332k</td><td>162k</td><td>[Ω]</td></tr> </table>	Vout _{set} =	15	24	[VDC]	R _{up} (E96) ≈	32k4	7k32	[Ω]	Vout _{set} =	10	9	[VDC]	R _{down} (E96) ≈	453k	267k	[Ω]	Vout _{set} =	20	24	[VDC]	R _{up} (E96) ≈	16k9	9k09	[Ω]	Vout _{set} =	12	9.99	[VDC]	R _{down} (E96) ≈	332k	162k	[Ω]
Vout _{set} =	5	[VDC]																																																																	
R _{up} (E96) ≈	57k6	[Ω]																																																																	
Vout _{set} =	2.5	1.8	1.5	1.1	[VDC]																																																														
R _{down} (E96) ≈	182k	52k3	26k7	3k48	[Ω]																																																														
Vout _{set} =	5.5	9	[VDC]																																																																
R _{up} (E96) ≈	205k	23k7	[Ω]																																																																
Vout _{set} =	3.3	2.5	[VDC]																																																																
R _{down} (E96) ≈	133k	59k	[Ω]																																																																
Vout _{set} =	15	24	[VDC]																																																																
R _{up} (E96) ≈	32k4	7k32	[Ω]																																																																
Vout _{set} =	10	9	[VDC]																																																																
R _{down} (E96) ≈	453k	267k	[Ω]																																																																
Vout _{set} =	20	24	[VDC]																																																																
R _{up} (E96) ≈	16k9	9k09	[Ω]																																																																
Vout _{set} =	12	9.99	[VDC]																																																																
R _{down} (E96) ≈	332k	162k	[Ω]																																																																

REGULATIONS		
Parameter	Condition	Value
Output Accuracy		±1% typ. / ±3% max.
Line Regulation	low line to high line, full load	0.25±% typ. / ±0.5% max.
Load Regulation	10% to 100% load	0.05% typ.
Transient Response	25% load step change recovery time	200mV 100µs

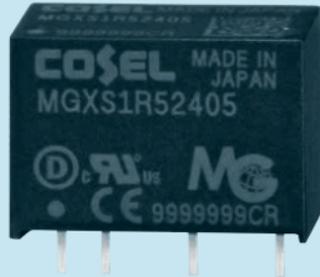
PROTECTIONS		
Parameter	Condition	Value
Short Circuit Protection (SCP)	less than 50mΩ	hiccup mode, automatic recovery
Over Current Protection (OCP)		120% min.
Over Temperature Protection (OTP)	case temperature (measured on tc point)	DC-DC OFF DC-DC ON
		105°C min., auto restart after cool down 100°C typ.

ENVIRONMENTAL		
Parameter	Condition	Value
Operating Temperature Range ⁽⁶⁾	@ natural convection 0.1m/s with derating (refer to <i>"Derating Graph"</i>)	-40°C to +100°C
Maximum Case Temperature	measured on tc point (refer to <i>"Dimension Drawing"</i>)	105°C
Temperature Coefficient		0.02%/°K
Thermal Impedance ⁽⁶⁾	0.1m/s, horizontal (T _{CASE} to T _{AMB})	12KW
Operating Altitude ⁽⁷⁾	with derating @ natural convection 0.1m/s	5000m
Operating Humidity	non-condensing	5% - 95% RH max.

continued on next page

MGXS1R5

MGX S 1R5 24 05 -□



- ① Series name
- ② Single output
- ③ Output wattage
- ④ Input voltage
- ⑤ Output voltage
- ⑥ Optional
- Y2: Output voltage adjustable (+10%, -5%)

MODEL	MGXS1R5243R3	MGXS1R52405	MGXS1R52412	MGXS1R52415
MAX OUTPUT WATTAGE[W]	1.32	1.50	1.56	1.50
DC OUTPUT	VOLTAGE[V]	3.3	5	12
	CURRENT[A]	0.4	0.3	0.13

SPECIFICATIONS

	MODEL	MGXS1R5243R3	MGXS1R52405	MGXS1R52412	MGXS1R52415	
INPUT	VOLTAGE[V]	DC6 - 60 (Surge voltage 76V, 100ms max) (Refer to the instruction manual for input voltage derating.)				
	CURRENT[A]	*1 0.072typ	0.080typ	0.080typ	0.077typ	
	EFFICIENCY[%]	*1 77typ	79typ	82typ	82typ	
OUTPUT	VOLTAGE[V]	3.3	5	12	15	
	CURRENT[A]	0.4	0.3	0.13	0.1	
	LINE REGULATION[mV]	20max	20max	48max	60max	
	LOAD REGULATION[mV]	20max	20max	48max	60max	
	RIPPLE[mVp-p]	*2 120max	120max	150max	150max	
	RIPPLE NOISE[mVp-p]	*2 200max	200max	200max	200max	
	TEMPERATURE REGULATION[mV]	-20 to +85°C	50max	50max	150max	180max
		-40 to +85°C	80max	80max	240max	290max
	DRIFT[mV]	*3 20max	20max	48max	60max	
	START-UP TIME[ms]	30max				
OUTPUT VOLTAGE SETTING[V]	3.21 - 3.42	4.90 - 5.21	11.64 - 12.36	14.55 - 15.45		
PROTECTION CIRCUIT	OVERCURRENT PROTECTION	Works over 105% of rating and recovers automatically				

GENERAL SPECIFICATIONS

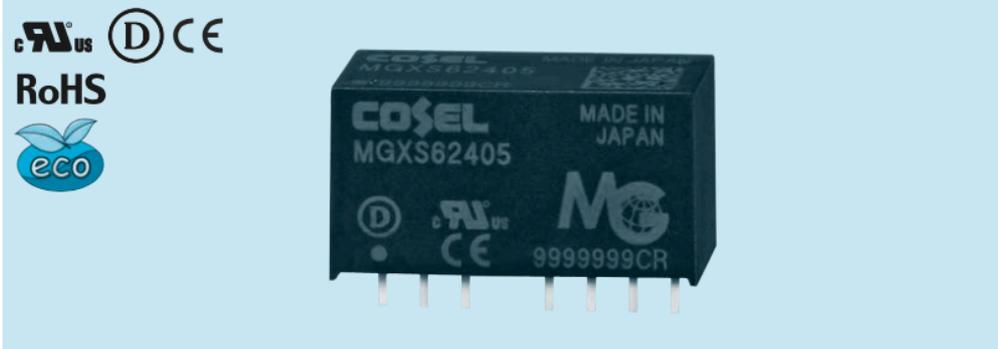
ISOLATION	INPUT-OUTPUT	DC1,500V or AC1,000V 1minute, Cutoff current=10mA, DC500V 1,000MΩ min (20±15°C)
ENVIRONMENT	OPERATING TEMP., HUMID. AND ALTITUDE	-40 to +85°C, 20 to 95%RH (Non condensing) (Required derating), 5,000m (16,400feet) max
	STORAGE TEMP., HUMID. AND ALTITUDE	-40 to +100°C, 20 to 95%RH (Non condensing), 9,000m (30,000feet) max
	VIBRATION	10 - 55Hz 98.0m/s ² (10G), 3minute period, 60minutes each along X, Y and Z axis
	IMPACT	490.3m/s ² (50G) 11ms, once each along X, Y and Z axis
SAFETY	AGENCY APPROVALS	UL60950-1, C-UL, EN60950-1
OTHERS	CASE SIZE/WEIGHT	17.0 X 12.0 X 8.5mm [0.67 X 0.48 X 0.34 inches] (W X H X D) / 4g max
	COOLING METHOD	Convection/Forced air

- *1 Rated input 24V DC Io=100%
- *2 Ripple and ripple noise is measured by using test board with ceramic capacitor 1μF at 50mm from output pins.
- *3 Drift is the DC output accuracy for eight hours period after a half-hour warm-up at 25°C.
- * Parallel operation with other model is not possible.
- * MGXW1R52412/MGXW1R52415 is available as single output, +24V/+30V

MGXS6

MGX S 6 24 05 -□

① ② ③ ④ ⑤ ⑥



- ① Series name
 - ② Single output
 - ③ Output wattage
 - ④ Input voltage
 - ⑤ Output voltage
 - ⑥ Optional
- Y2: Output voltage adjustable (+10%, -5%)

MODEL	MGXS6243R3	MGXS62405	MGXS62412	MGXS62415
MAX OUTPUT WATTAGE[W]	5.28	6.0	6.0	6.0
DC OUTPUT	VOLTAGE[V]	3.3	5	12
	CURRENT[A]	1.6	1.2	0.5

SPECIFICATIONS

	MODEL	MGXS6243R3	MGXS62405	MGXS62412	MGXS62415	
INPUT	VOLTAGE[V]	DC6 - 60 (Surge voltage 76V, 100ms max) (Refer to the instruction manual for input voltage derating.)				
	CURRENT[A]	*1 0.26typ	0.29typ	0.29typ	0.29typ	
	EFFICIENCY[%]	*1 85typ	88typ	87typ	87typ	
OUTPUT	VOLTAGE[V]	3.3	5	12	15	
	CURRENT[A]	1.6	1.2	0.5	0.4	
	LINE REGULATION[mV]	20max	20max	48max	60max	
	LOAD REGULATION[mV]	20max	20max	48max	60max	
	RIPPLE[mVp-p]	Io=30% -	75max	75max	100max	100max
		Io=0 - 30%	225max	225max	300max	300max
		Vin=DC48 - 60V	225max	225max	300max	300max
	RIPPLE NOISE[mVp-p]	Io=30% -	120max	120max	150max	150max
		Io=0 - 30%	300max	300max	400max	400max
		Vin=DC48 - 60V	300max	300max	400max	400max
	TEMPERATURE REGULATION[mV]	-20 to +75°C	50max	50max	150max	180max
	-40 to +75°C	80max	80max	240max	290max	
DRIFT[mV]	*3 20max	20max	48max	60max		
START-UP TIME[ms]	30max					
OUTPUT VOLTAGE SETTING[V]	3.21 - 3.42	4.90 - 5.21	11.64 - 12.36	14.55 - 15.45		
PROTECTION CIRCUIT AND OTHERS	OVERCURRENT PROTECTION	Works over 105% of rating and recovers automatically				
	REMOTE ON/OFF	Provided (Negative logic L : ON, H : OFF)				

GENERAL SPECIFICATIONS

ISOLATION	INPUT-OUTPUT	DC1,500V or AC1,000V 1minute, Cutoff current=10mA, DC500V 1,000MΩ min (20±15°C)
ENVIRONMENT	OPERATING TEMP., HUMID. AND ALTITUDE	-40 to +85°C, 20 to 95%RH (Non condensing) (Required derating), 5,000m (16,400feet) max
	STORAGE TEMP., HUMID. AND ALTITUDE	-40 to +100°C, 20 to 95%RH (Non condensing), 9,000m (30,000feet) max
	VIBRATION	10 - 55Hz 98.0m/s ² (10G), 3minute period, 60minutes each along X, Y and Z axis
	IMPACT	490.3m/s ² (50G) 11ms, once each along X, Y and Z axis
SAFETY	AGENCY APPROVALS	UL60950-1, C-UL, EN60950-1
OTHERS	CASE SIZE/WEIGHT	22.0 × 12.0 × 9.5mm [0.87 × 0.48 × 0.38 inches] (W × H × D) / 7g max
	COOLING METHOD	Convection/Forced air

- *1 Rated input 24V DC Io=100%
- *2 Ripple and ripple noise is measured by using test board with ceramic capacitor 1μF at 50mm from output pins.
- *3 Drift is the DC output accuracy for eight hours period after a half-hour warm-up at 25°C.
- * Parallel operation with other model is not possible.
- * MGXW62412/MGXW62415 is available as single output, +24V/+30V

MAX31820PAR

1-Wire, Parasite-Power, Ambient Temperature Sensor

General Description

The MAX31820PAR ambient temperature sensor provides 9-bit to 12-bit Celsius temperature measurements with $\pm 0.5^{\circ}\text{C}$ accuracy over a $+10^{\circ}\text{C}$ to $+45^{\circ}\text{C}$ temperature range. Over its entire -55°C to $+125^{\circ}\text{C}$ operating range, the device has $\pm 2.0^{\circ}\text{C}$ accuracy.

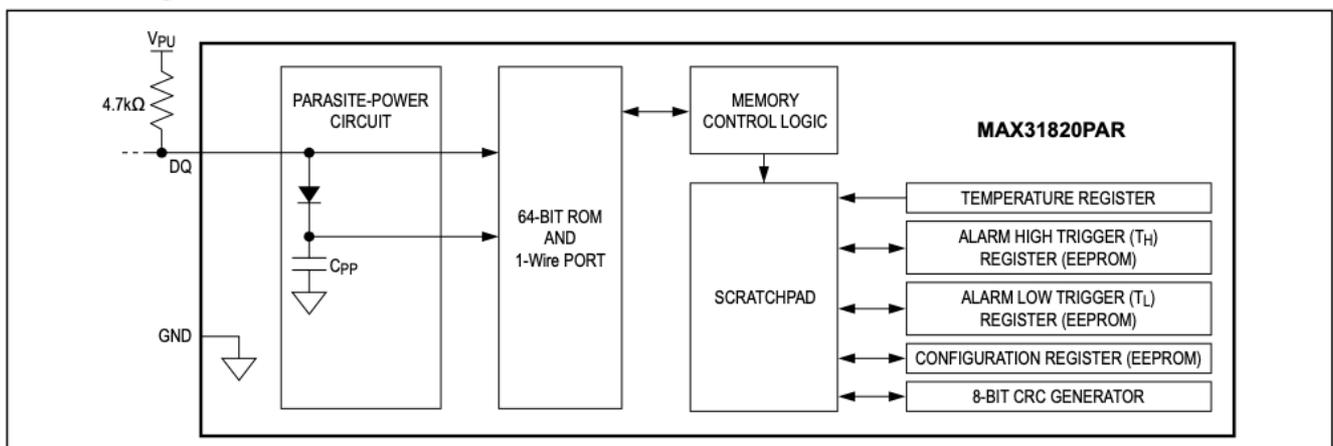
The device communicates over a 1-Wire[®] bus that, by definition, requires only one data line (and ground) for communication with a central microprocessor. In addition, the device derives power directly from the data line ("parasite power"), eliminating the need for an external power supply. Requiring so few pins enables the device to be placed in a 3-pin TO-92 package. The form factor of this package allows the device to be placed above the board and thus measure the ambient temperature of a system, as opposed to the board temperature that a surface-mount package would measure.

Each MAX31820PAR has a unique 64-bit serial code, which allows multiple MAX31820PAR devices to function on the same 1-Wire bus. Therefore, it is simple to use one microprocessor to control many devices distributed over a large area.

Applications

- HVAC Environmental Controls
- Temperature Monitoring Systems Inside Buildings, Equipment, or Machinery
- Process Monitoring and Control Systems
- Thermostatic Controls
- Industrial Systems
- Consumer Products
- Thermometers
- Any Thermally Sensitive System

Block Diagram



1-Wire is a registered trademark of Maxim Integrated Products, Inc.

Benefits and Features

- Unique 1-Wire Interface Requires Only One Port Pin for Communication
- Derives Power from Data Line (Parasite Power); No Local Power Supply Needed
- Multidrop Capability Simplifies Distributed Temperature-Sensing Applications
- Requires No External Components
- Measures Temperatures from -55°C to $+125^{\circ}\text{C}$ (-67°F to $+257^{\circ}\text{F}$)
- $\pm 0.5^{\circ}\text{C}$ Accuracy from $+10^{\circ}\text{C}$ to $+45^{\circ}\text{C}$
- Thermometer Resolution is User-Selectable from 9 Bits to 12 Bits
- Converts Temperature to 12-Bit Digital Word in 750ms (Max)
- User-Definable Nonvolatile (NV) Alarm Settings
- Alarm Search Command Identifies and Addresses Devices Whose Temperature is Outside Programmed Limits (Temperature Alarm Condition)
- Available in 3-Pin TO-92 Package
- Software Compatible with the DS1822-PAR and DS18B20-PAR

Ordering Information appears at end of data sheet.

For related parts and recommended products to use with this part, refer to www.maximintegrated.com/MAX31820PAR.related.

Absolute Maximum Ratings

Voltage Range on Any Pin Relative to Ground-0.5V to +6.0V Storage Temperature Range -55°C to +125°C
 Operating Temperature Range..... -55°C to +100°C Soldering Temperature (reflow)..... +260°C

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Electrical Characteristics

(V_{PU} = 3.0V to 3.7V, T_A = -55°C to +100°C, unless otherwise noted.) (Note 1)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Pullup Supply Voltage	V _{PU}	(Notes 2, 3)	3.0		3.7	V
Thermometer Error	T _{ERR}	+10°C to +45°C			±0.5	°C
		-55°C to +100°C			±2	
Input Logic-Low	V _{IL}	(Notes 2, 4, 5)	-0.3		+0.8	V
Input Logic-High	V _{IH}	(Notes 2, 6)	3.0		3.7	V
Sink Current	I _L	V _{I/O} = 0.4V (Note 2)	4.0			mA
Active Current	I _{DQA}	(Note 7)		1	1.5	mA
DQ Input Current	I _{DQ}	(Note 8)		5		µA
Drift		(Note 9)		±0.2		°C

1. Feature List and Ordering Information

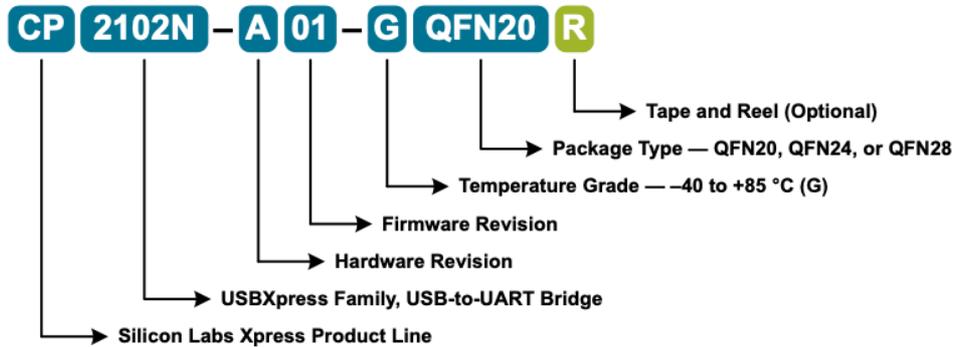


Figure 1.1. CP2102N Part Numbering

The CP2102N devices have the following features:

- **Single-Chip USB-to-UART Data Transfer**
 - Integrated USB transceiver; no external resistors required
 - Integrated clock; no external crystal required
 - Internal 960-byte programmable ROM for vendor ID, product ID, serial number, power descriptor, release number, and product description strings
 - On-chip power-on reset circuit
 - On-chip voltage regulator — 3.3 V output
 - Pin compatible with CP2101/2/9 (QFN28 package)
 - Pin compatible with CP2104 (QFN24 package)
- **USB Function Controller**
 - USB Specification 2.0 compliant; full-speed (12 Mbps)
 - USB suspend states supported via SUSPEND pins
 - USB Battery Charger Detection (USB BCS 1.2 Specification)
 - Remote wakeup for waking a suspended host
- **Single power supply of 3.0 to 3.6 V or 3.0 to 5.25 V**
- **Universal Asynchronous Receiver/Transmitter (UART)**
 - All handshaking and modem interface signals
 - Data formats supported
 - Data bits — 5, 6, 7, and 8
 - Stop bits — 1, 1.5, and 2
 - Parity — odd, even, mark, space, no parity
 - Baud rates: 300 baud to 3 Mbaud
 - 512 byte receive buffer
 - 512 byte transmit buffer
 - Hardware or Xon/Xoff handshaking supported
- **Virtual COM Port Device Drivers**
 - Works with existing COM port Applications
 - Supported on Windows, Mac, and Linux
 - Royalty-free distribution license
- **Direct Driver Support**
 - Royalty-free distribution license

Table 1.1. Product Selection Guide

Ordering Part Number	GPIOs	Battery Charger Detect	Separate VIO and VDD Pins	Pb-free (RoHS Compliant)	Temperature Range	Package
CP2102N-A02-GQFN28	7	Yes	—	Yes	-40 to +85 °C	QFN28
CP2102N-A02-GQFN24	4	—	Yes	Yes	-40 to +85 °C	QFN24
CP2102N-A02-GQFN20	4	—	—	Yes	-40 to +85 °C	QFN20

Note:

1. Devices with the same ordering part number may have different types of pin 1 indicators. However, all of these variants can use the same landing diagram as long as the recommended landing diagram instructions are followed.

2.2 Battery Charger Detect

The CP2102N Battery Charger Detect notifies an external battery charger the amount of current available from the USB interface.

The figure below shows an example connection diagram for external battery charging circuitry. If using an external battery charging IC, consult the data sheet for more information about the specific recommended connection diagrams.

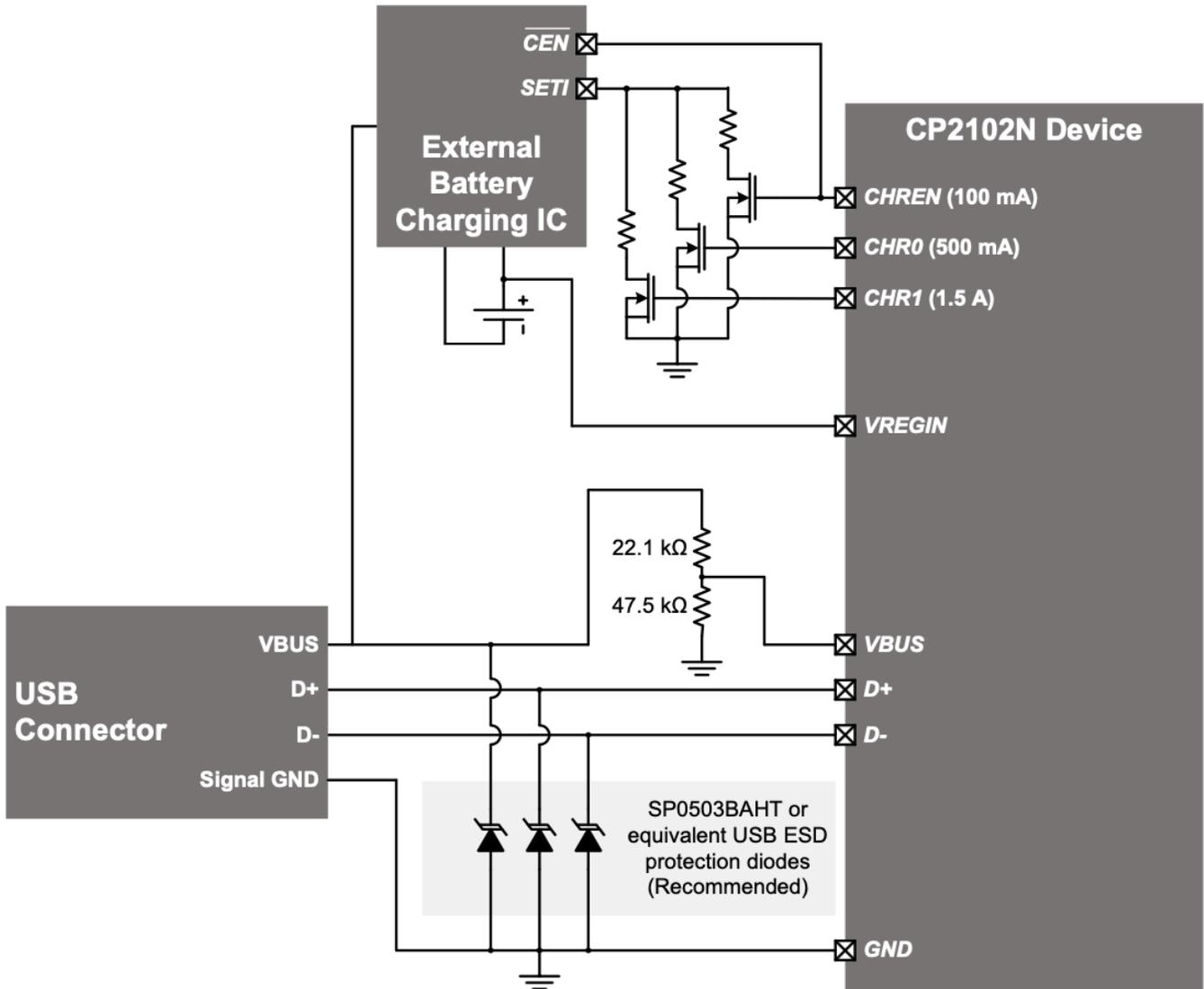


Figure 2.4. Battery Charging Connection Diagram

3. Electrical Specifications

3.1 Electrical Characteristics

All electrical parameters in all tables are specified under the conditions listed in [Table 3.1 Recommended Operating Conditions](#) on page 10, unless stated otherwise.

3.1.1 Recommended Operating Conditions

Table 3.1. Recommended Operating Conditions

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Operating Supply Voltage on VDD ¹	V _{DD}		3.0	—	3.6	V
Operating Supply Voltage on VIO ³	V _{IO}		1.71	—	V _{DD}	V
Operating Supply Voltage on VREGIN	V _{REGIN}		3.0	—	5.25	V
Operating Ambient Temperature	T _A		-40	—	85	°C

Note:

1. Standard USB compliance tests require 3.0 V on VDD for compliant operation.
2. All voltages with respect to GND.
3. On devices without a VIO pin, V_{IO} = V_{DD}.
4. GPIO levels are undefined whenever VIO is less than 1 V.

3.1.2 Power Consumption

Table 3.2. Power Consumption

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
Normal Operation ^{1, 2}	I _{DD}	115200 baud transmitting continuous bidirectional data	—	9.5	—	mA
		3 Mbaud transmitting continuous bidirectional data	—	13.7	—	mA
USB Suspend ^{1, 2}	I _{DD}		—	195	—	μA
Held in Reset ^{1, 2}	I _{DD}		—	1.3	—	mA
USB Pull-up ³	I _{PU}		—	200	230	μA

Note:

1. Includes supply current from internal LDO regulator, supply monitor, and internal oscillators. These power consumption numbers are only for the CP2102N and do not include an external RS232 transceiver or other external circuitry.
2. USB Pull-up current should be added for total supply current. Normal and suspended supply current is current flowing into VREGIN.
3. The USB Pull-up supply current values are calculated values based on USB specifications. USB Pull-up supply current is current flowing from VDD to GND through USB pull-down/pull-up resistors on D+ and D-.

BS170

Small Signal MOSFET 500 mA, 60 Volts N-Channel TO-92 (TO-226)

Features

- This is a Pb-Free Device*

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Drain-Source Voltage	V_{DS}	60	Vdc
Gate-Source Voltage - Continuous - Non-repetitive ($t_p \leq 50 \mu s$)	V_{GS} V_{GSM}	± 20 ± 40	Vdc Vpk
Drain Current (Note)	I_D	0.5	Adc
Total Device Dissipation @ $T_A = 25^\circ C$	P_D	350	mW
Operating and Storage Junction Temperature Range	T_J, T_{stg}	-55 to +150	$^\circ C$

Stresses exceeding those listed in the Maximum Ratings table may damage the device. If any of these limits are exceeded, device functionality should not be assumed, damage may occur and reliability may be affected.

NOTE: The Power Dissipation of the package may result in a lower continuous drain current.

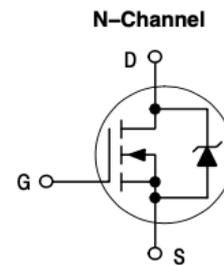
*For additional information on our Pb-Free strategy and soldering details, please download the ON Semiconductor Soldering and Mounting Techniques Reference Manual, SOLDERRM/D.



ON Semiconductor®

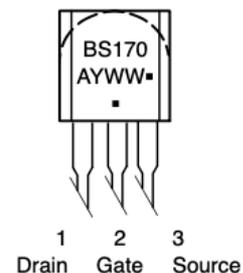
www.onsemi.com

500 mA, 60 Volts
 $R_{DS(on)} = 5.0 \Omega$



TO-92 (TO-226)
CASE 29
STYLE 30

MARKING DIAGRAM & PIN ASSIGNMENT



A = Assembly Location
Y = Year
WW = Work Week
▪ = Pb-Free Package

(Note: Microdot may be in either location)

ORDERING INFORMATION

See detailed ordering and shipping information in the package dimensions section on page 2 of this data sheet.

BS170

ELECTRICAL CHARACTERISTICS ($T_A = 25^\circ\text{C}$ unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
----------------	--------	-----	-----	-----	------

OFF CHARACTERISTICS

Gate Reverse Current ($V_{GS} = 15\text{ Vdc}$, $V_{DS} = 0$)	I_{GSS}	–	0.01	10	nAdc
Drain–Source Breakdown Voltage ($V_{GS} = 0$, $I_D = 100\ \mu\text{Adc}$)	$V_{(BR)DSS}$	60	90	–	Vdc

ON CHARACTERISTICS (Note 1)

Gate Threshold Voltage ($V_{DS} = V_{GS}$, $I_D = 1.0\ \text{mAdc}$)	$V_{GS(Th)}$	0.8	2.0	3.0	Vdc
Static Drain–Source On Resistance ($V_{GS} = 10\ \text{Vdc}$, $I_D = 200\ \text{mAdc}$)	$r_{DS(on)}$	–	1.8	5.0	Ω
Drain Cutoff Current ($V_{DS} = 25\ \text{Vdc}$, $V_{GS} = 0\ \text{Vdc}$)	$I_{D(off)}$	–	–	0.5	μA
Forward Transconductance ($V_{DS} = 10\ \text{Vdc}$, $I_D = 250\ \text{mAdc}$)	g_{fs}	–	200	–	mmhos

SMALL–SIGNAL CHARACTERISTICS

Input Capacitance ($V_{DS} = 10\ \text{Vdc}$, $V_{GS} = 0$, $f = 1.0\ \text{MHz}$)	C_{iss}	–	–	60	pF
---	-----------	---	---	----	----

SWITCHING CHARACTERISTICS

Turn–On Time ($I_D = 0.2\ \text{Adc}$) See Figure 1	t_{on}	–	4.0	10	ns
Turn–Off Time ($I_D = 0.2\ \text{Adc}$) See Figure 1	t_{off}	–	4.0	10	ns

Product parametric performance is indicated in the Electrical Characteristics for the listed test conditions, unless otherwise noted. Product performance may not be indicated by the Electrical Characteristics if operated under different conditions.

1. Pulse Test: Pulse Width $\leq 300\ \mu\text{s}$, Duty Cycle $\leq 2.0\%$.

ORDERING INFORMATION

Device	Package	Shipping†
BS170	TO–92 (TO–226) (Pb–Free)	1000 Unit/Tube
BS170RLRAG	TO–92 (TO–226) (Pb–Free)	2000 Tape & Reel

†For information on tape and reel specifications, including part orientation and tape sizes, please refer to our Tape and Reel Packaging Specifications Brochure, BRD8011/D.

BS170

RESISTIVE SWITCHING

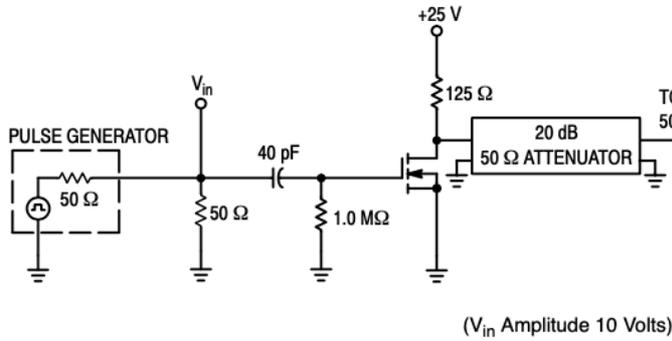


Figure 1. Switching Test Circuit

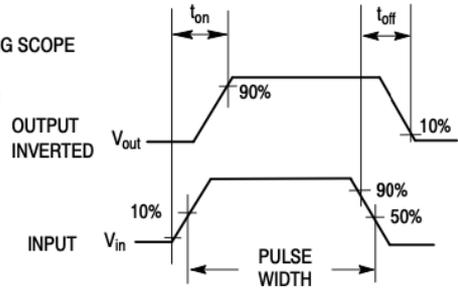


Figure 2. Switching Waveforms

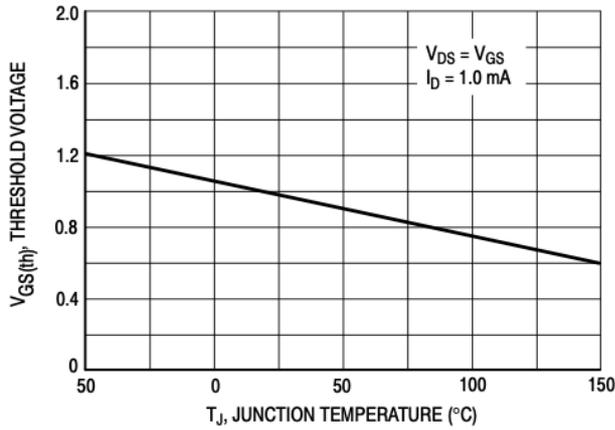


Figure 3. $V_{GS(th)}$ Normalized versus Temperature

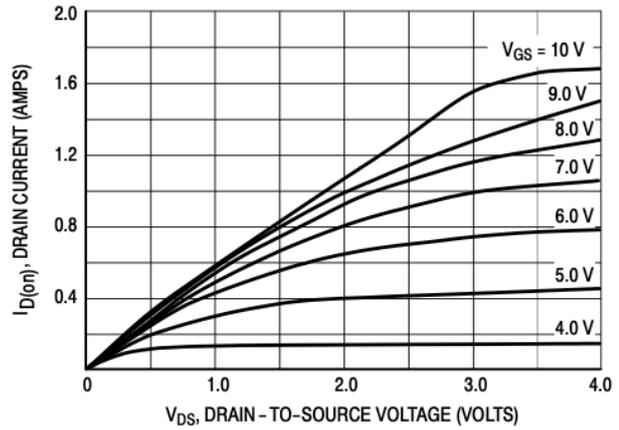


Figure 4. On-Region Characteristics

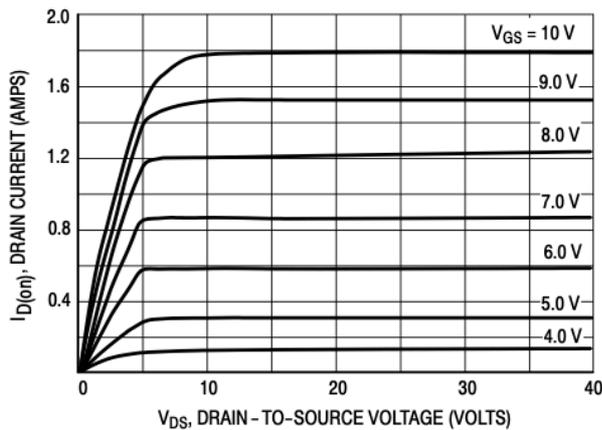


Figure 5. Output Characteristics

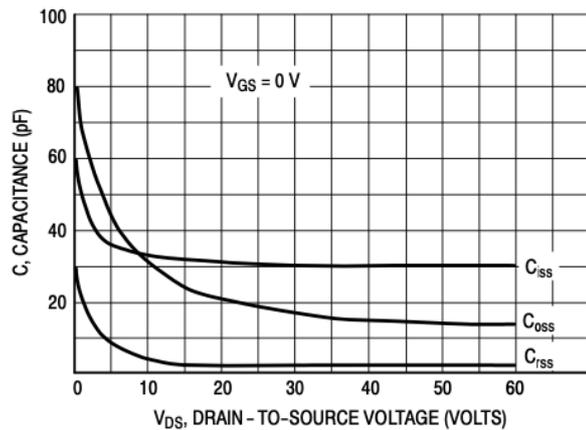


Figure 6. Capacitance versus Drain-To-Source Voltage

N-channel 80 V, 5.0 mΩ typ., 80 A STripFET™ F7 Power MOSFET in a TO-220 package

Datasheet - production data

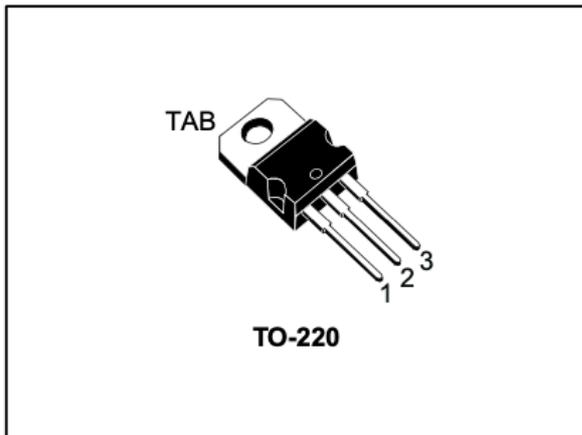
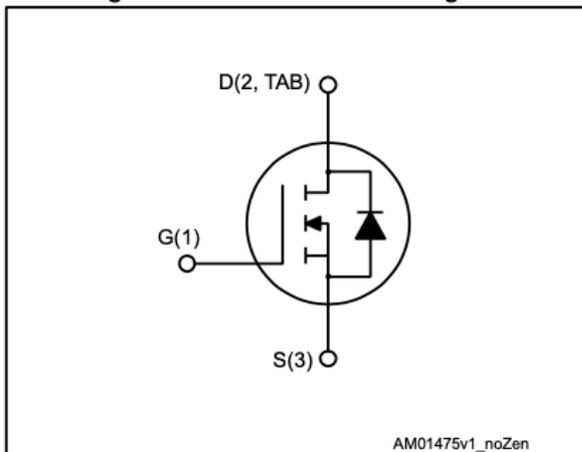


Figure 1: Internal schematic diagram



Features

Order code	V _{DS}	R _{DS(on)} max.	I _D	P _{TOT}
STP130N8F7	80 V	5.8 mΩ	80 A	205 W

- Among the lowest R_{DS(on)} on the market
- Excellent FoM (figure of merit)
- Low C_{rss}/C_{iss} ratio for EMI immunity
- High avalanche ruggedness

Applications

- Switching applications

Description

This N-channel Power MOSFET utilizes STripFET™ F7 technology with an enhanced trench gate structure that results in very low on-state resistance, while also reducing internal capacitance and gate charge for faster and more efficient switching.

Table 1: Device summary

Order code	Marking	Package	Packing
STP130N8F7	130N8F7	TO-220	Tube

1 Electrical ratings

Table 2: Absolute maximum ratings

Symbol	Parameter	Value	Unit
V_{DS}	Drain-source voltage	80	V
V_{GS}	Gate-source voltage	± 20	V
I_D	Drain current (continuous) at $T_C = 25\text{ }^\circ\text{C}$	80	A
I_D	Drain current (continuous) at $T_C = 100\text{ }^\circ\text{C}$	80	A
$I_{DM}^{(1)}$	Drain current (pulsed)	320	A
P_{TOT}	Total dissipation at $T_C = 25\text{ }^\circ\text{C}$	205	W
$E_{AS}^{(2)}$	Single pulse avalanche energy	320	mJ
T_j	Operating junction temperature range	-55 to 175	$^\circ\text{C}$
T_{stg}	Storage temperature range		

Notes:

(1)Pulse width is limited by safe operating area

(2)Starting $T_j = 25\text{ }^\circ\text{C}$, $I_D = 40\text{ A}$, $V_{DD} = 40\text{ V}$

Table 3: Thermal data

Symbol	Parameter	Value	Unit
$R_{th\text{-case}}$	Thermal resistance junction-case	0.73	$^\circ\text{C/W}$
$R_{th\text{-amb}}$	Thermal resistance junction-ambient	62.5	$^\circ\text{C/W}$

2 Electrical characteristics

(T_{CASE} = 25 °C unless otherwise specified)

Table 4: On/off states

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
V _{(BR)DSS}	Drain-source breakdown voltage	V _{GS} = 0 V, I _D = 250 μA	80			V
I _{DSS}	Zero gate voltage drain current	V _{GS} = 0 V, V _{DS} = 80 V			1	μA
		V _{GS} = 0 V, V _{DS} = 80 V, T _J = 125 °C ⁽¹⁾			100	μA
I _{GSS}	Gate-source leakage current	V _{DS} = 0 V, V _{GS} = 20 V			100	nA
V _{GS(th)}	Gate threshold voltage	V _{DS} = V _{GS} , I _D = 250 μA	2.5		4.5	V
R _{DS(on)}	Static drain-source on-resistance	V _{GS} = 10 V, I _D = 40 A		5.0	5.8	mΩ

Notes:

⁽¹⁾Defined by design, not subject to production test.

Table 5: Dynamic

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
C _{iss}	Input capacitance	V _{DS} = 25 V, f = 1 MHz, V _{GS} = 0 V	-	4500	-	pF
C _{oss}	Output capacitance		-	1100	-	pF
C _{rss}	Reverse transfer capacitance		-	110	-	pF
Q _g	Total gate charge	V _{DD} = 40 V, I _D = 80 A, V _{GS} = 0 to 10 V (see Figure 14: "Test circuit for gate charge behavior")	-	60	-	nC
Q _{gs}	Gate-source charge		-	25	-	nC
Q _{gd}	Gate-drain charge		-	15	-	nC

Table 6: Switching times

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
t _{d(on)}	Turn-on delay time	V _{DD} = 40 V, I _D = 40 A, R _G = 4.7 Ω, V _{GS} = 10 V (see Figure 13: "Test circuit for resistive load switching times" and Figure 18: "Switching time waveform")	-	140	-	ns
t _r	Rise time		-	210	-	ns
t _{d(off)}	Turn-off-delay time		-	190	-	ns
t _f	Fall time		-	120	-	ns

4A, 200V - 600V Ultrafast Glass Passivated Rectifiers

FEATURES

- Ideally suited for use in very high frequency switching power supplies, inverters and as free wheeling diodes
- Ultrafast recovery time for high efficiency
- Excellent high temperature switching
- Glass passivated junction
- Compliant to RoHS Directive 2011/65/EU and in accordance to WEEE 2002/96/EC
- Halogen-free according to IEC 61249-2-21



DO-201AD

MECHANICAL DATA

Case: DO-201AD

Molding compound, UL flammability classification rating 94V-0

Part no. with suffix "H" means AEC-Q101 qualified

Packing code with suffix "G" means green compound (halogen-free)

Terminal: Pure tin plated leads, solderable per JESD22-B102

Meet JESD 201 class 2 whisker test

Weight: 1.2 g (approximately)

MAXIMUM RATINGS AND ELECTRICAL CHARACTERISTICS (T_A=25°C unless otherwise noted)

PARAMETER	SYMBOL	MUR420	MUR440	MUR460	UNIT
Maximum repetitive peak reverse voltage	V _{RRM}	200	400	600	V
Maximum RMS voltage	V _{RMS}	140	280	420	V
Maximum DC blocking voltage	V _{DC}	200	400	600	V
Maximum average forward rectified current	I _{F(AV)}	4			A
Peak forward surge current, 8.3 ms single half sine-wave superimposed on rated load	I _{FSM}	125			A
Maximum instantaneous forward voltage (Note 1) @ 4 A	V _F	0.89	1.28		V
Maximum reverse current @ rated V _R	I _R	5 150	10 250		μA
		T _J =25°C			
		T _J =125°C			
Maximum reverse recovery time (Note 2)	t _{rr}	25	50		ns
Typical junction capacitance (Note 3)	C _J	65			pF
Typical thermal resistance	R _{θJL}	15			°C/W
	R _{θJA}	28			
Operating junction temperature range	T _J	- 55 to +175			°C
Storage temperature range	T _{STG}	- 55 to +175			°C

Note 1: Pulse Test with PW=300μs, 1% Duty Cycle

Note 2: Reverse Recovery Test Conditions: I_F=0.5A, I_R=1.0A, I_{RR}=0.25A

Note 3: Measured at 1 MHz and Applied Reverse Voltage of 4.0V D.C.

ORDERING INFORMATION					
PART NO.	PART NO. SUFFIX	PACKING CODE	PACKING CODE SUFFIX (*)	PACKAGE	PACKING
MUR4x0 (Note 1)	H	A0	G	DO-201AD	500 / Ammo box
		R0		DO-201AD	1,250 / 13" Paper reel
		B0		DO-201AD	500 / Bulk packing
		X0		DO-201AD	Forming

Note 1: "x" defines voltage from 200V (MUR420) to 600V (MUR460)

*: Optional available

EXAMPLE					
PREFERRED P/N	PART NO.	PART NO. SUFFIX	PACKING CODE	PACKING CODE SUFFIX	DESCRIPTION
MUR460HA0G	MUR460	H	A0	G	AEC-Q101 qualified Green compound

RATINGS AND CHARACTERISTICS CURVES

(T_A=25°C unless otherwise noted)

FIG. 1- MAXIMUM FORWARD CURRENT DERATING CURVE

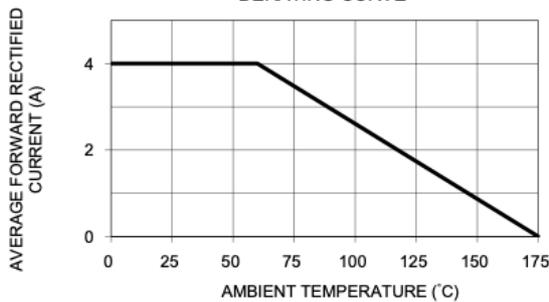


FIG. 2- MAXIMUM NON-REPETITIVE FORWARD SURGE CURRENT

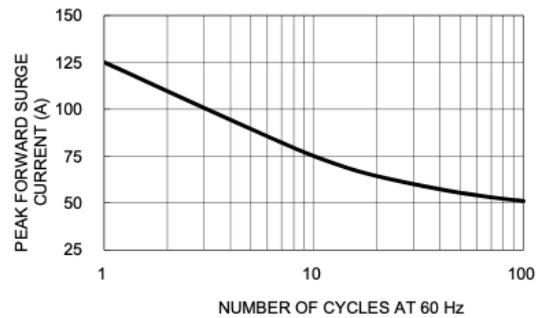


FIG. 3 TYPICAL INSTANTANEOUS FORWARD CHARACTERISTICS

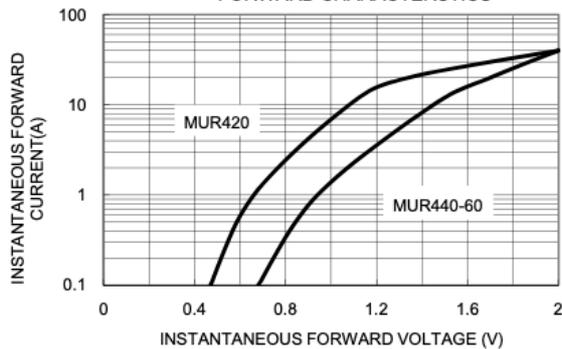
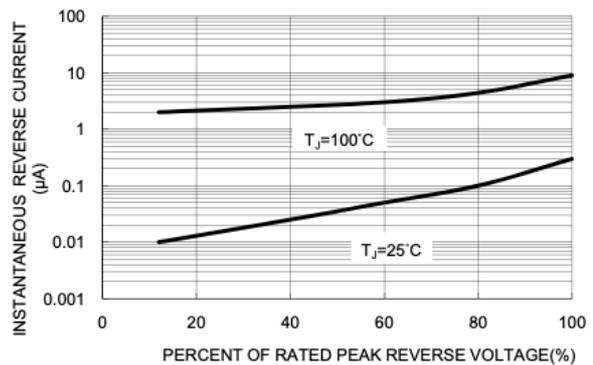


FIG. 4 TYPICAL REVERSE CHARACTERISTICS



2 Watts

- Single Output
- SMD Package
- Industry Standard Pinout
- Operating Temperature -40 °C to +105 °C
- 1500 VDC Isolation, 3000 VDC Option
- 3 Year Warranty



Dimensions:

ISH:
0.500 x 0.44 x 0.285" (12.7 x 11.2 x 7.25 mm)

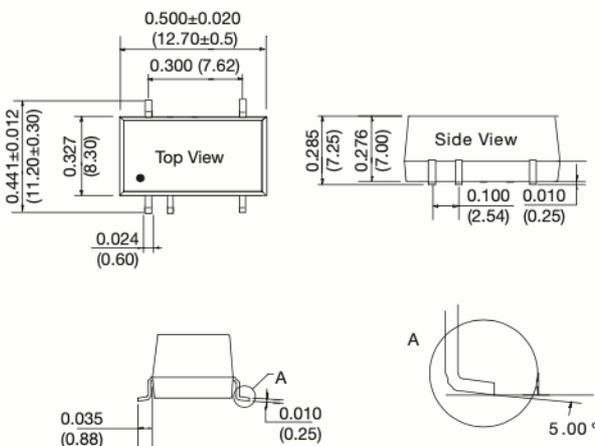
Models & Ratings

Input Voltage	Output Voltage	Output Current	Input Current ⁽²⁾		Maximum Capacitive Load	Efficiency ⁽³⁾	Model Number ^(1,4)
			No Load	Full Load			
4.5-5.5	3.3V	400 mA	30 mA	506 mA	220 µF	72%	ISH0503A ⁽¹⁾
	5V	400 mA	30 mA	506 mA	220 µF	79%	ISH0505A ⁽¹⁾
	9V	222 mA	30 mA	506 mA	220 µF	82%	ISH0509A ⁽¹⁾
	12V	167 mA	30 mA	506 mA	220 µF	82%	ISH0512A ⁽¹⁾
	15V	133 mA	30 mA	506 mA	220 µF	83%	ISH0515A ⁽¹⁾
10.8-13.2	5V	400 mA	25 mA	212 mA	220 µF	79%	ISH1205A ⁽¹⁾
	9V	222 mA	25 mA	212 mA	220 µF	82%	ISH1209A
	12V	167 mA	25 mA	212 mA	220 µF	82%	ISH1212A ⁽¹⁾
	15V	133 mA	25 mA	212 mA	220 µF	83%	ISH1215A ⁽¹⁾
	24V	83 mA	25 mA	212 mA	220 µF	84%	ISH1224A ⁽¹⁾
13.5-16.5	15V	133 mA	18 mA	169 mA	220 µF	83%	ISH1515A ⁽¹⁾
21.6-26.4	5V	400 mA	15 mA	105 mA	220 µF	79%	ISH2405A ⁽¹⁾
	12V	167 mA	15 mA	105 mA	220 µF	82%	ISH2412A ⁽¹⁾
	15V	133 mA	15 mA	105 mA	220 µF	83%	ISH2415A ⁽¹⁾
	24V	83 mA	15 mA	105 mA	220 µF	86%	ISH2424A ⁽¹⁾

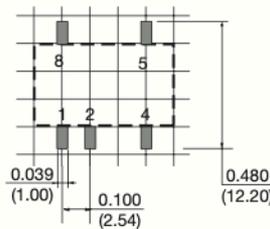
Notes

1. For optional 3000 VDC isolation add suffix '-H' to end of part number e.g. ISH1224A-H.
2. Input currents measured at nominal input voltage.
3. Typical value at full load.
4. For optional tape & reel option add suffix -TR to the end of the p/n. Reel quantity = 500

Mechanical Details



Recommended Footprint
Top View grid: 0.1 x 0.1 in (2.54 x 2.54 mm)



PIN CONNECTIONS	
Pin	Function
1	GND
2	+Vin
4	0V
5	+Vout
8	No Connection

Notes

1. All dimensions are in inches (mm)
2. Weight: 0.003 lbs (1.5 g) typical.
3. Pin diameter: 0.02 ±0.002 (0.5 ±0.005)
4. Pin pitch and length tolerance: ±0.014 (±0.35)
5. Case tolerance: ±0.02 (±0.5)

Input

Characteristic	Minimum	Typical	Maximum	Units	Notes & Conditions
Input Voltage Range	4.50		5.50	VDC	5 V nominal
	10.80		13.20	VDC	12 V nominal
	13.50		16.50	VDC	15 V nominal
	21.60		26.40	VDC	24 V nominal
Input Current					See Models and Ratings table
Input Reflected Ripple		15		mA pk-pk	Through 12 μ H inductor and 47 μ F capacitor
Input Surge			9	VDC for 1 s	5 V models
			18	VDC for 1 s	12 V models
			21	VDC for 1 s	15 V models
			30	VDC for 1 s	24 V models
Input Filter	Capacitor				

Output

Characteristic	Minimum	Typical	Maximum	Units	Notes & Conditions
Output Voltage	3.3		24	VDC	See Models and Ratings table
Initial Set Accuracy			-7.5, +2.5	%	At 70% load
Minimum Load	10			%	
Line Regulation			± 1.2	%	Per 1% change of input voltage ($\pm 1.5\%$ for 3V3 output)
Load Regulation				%	See graph
Start Up Delay		2		ms	
Ripple and Noise		100		mV pk-pk	20 MHz bandwidth, measured using 0.1 μ F capacitor
Transient Response			3	% deviation	Recovery to within 1% in 500 μ s for a 25% load change (5% max. deviation for 3.3 & 5V models)
Short Circuit Protection					Continuous, with auto recovery, except 1 s max for 24 input V models
Maximum Capacitive Load			220	μ F	
Temperature Coefficient			0.03	%/°C	

General

Characteristic	Minimum	Typical	Maximum	Units	Notes & Conditions
Efficiency					See Models and Ratings table
Isolation: Input to Output	1500			VDC	For optional high isolation versions, 3000 VDC input to output add suffix -H to model number
Switching Frequency	100	115	135	kHz	
Isolation Resistance	10^9			Ω	Input to output, tested at 500 VDC
Isolation Capacitance		20		pF	Input to output
Power Density			31.8	W/in ²	
Mean Time Between Failure	3500			kHrs	MIL-HDBK-217F, +25 °C GB
Weight		0.003 (1.5)		lb (g)	
Recommended Solder Profile	IPC/JEDEC J-STD-020D.1				

Environmental

Characteristic	Minimum	Typical	Maximum	Units	Notes & Conditions
Operating Temperature	-40		+105	°C	Derate from 100% load at +100 °C to 80% load at 105 °C
Storage Temperature	-55		+125	°C	
Case Temperature			+115	°C	
Operating Humidity			95	% RH	Non-condensing
Cooling					Natural convection

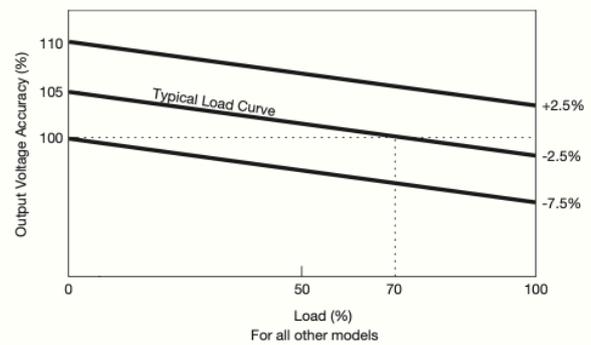
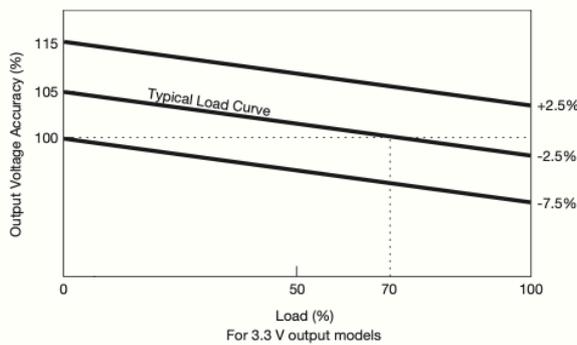
EMC: Emissions

Phenomenon	Standard	Test Level	Notes & Conditions
Conducted	EN55022	Class B	See Application Note for Class B filter
Radiated	EN55022	Class B	See Application Note for Class B filter

EMC: Immunity

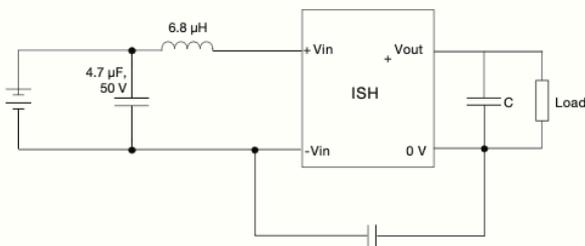
Phenomenon	Standard	Test Level	Criteria	Notes & Conditions
ESD Immunity	EN61000-4-2	3	B	
Radiated Immunity	EN61000-4-3	3 V/m	A	
EFT/Burst	EN61000-4-4	2	B	External input capacitor required, 330 μ F/100 V
Surge	EN61000-4-5	2	B	External input capacitor required, 330 μ F/100 V
Conducted Immunity	EN61000-4-6	3 V rms	A	
Magnetic Fields	EN61000-4-8	1 A/m	A	

Load Regulation



Application Note

EMI Filter for Class B Emissions

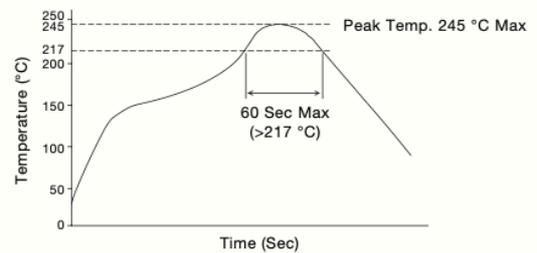


For ISH2424A model only, add Ycap, 470 pF, 3 kv

Output Voltage	C
3.3	10.00 μ F
5	10.00 μ F
9	4.70 μ F
12	2.20 μ F
15	1.00 μ F
24	0.47 μ F

Solder Profile

It is recommended to refer to IPC/JEDEC J-STD-020D standard for reflow soldering curve. The recommended reflow soldering temperature graph for our products is as follows:



Note:

The curve is only suitable for hot air convection reflow soldering.

Thermally Enhanced, Fully Integrated, Hall-Effect-Based Linear Current Sensor IC with 100 $\mu\Omega$ Current Conductor

FEATURES AND BENEFITS

- Industry-leading noise performance through proprietary amplifier and filter design techniques
- Integrated shield greatly reduces capacitive coupling from current conductor to die due to high dV/dt signals, and prevents offset drift in high-side, high-voltage applications
- Total output error improvement through gain and offset trim over temperature
- Small package size, with easy mounting capability
- Monolithic Hall IC for high reliability
- Ultralow power loss: 100 $\mu\Omega$ internal conductor resistance
- Galvanic isolation allows use in economical, high-side current sensing in high-voltage systems
- AEC-Q100 qualified

Continued on the next page...



PACKAGE: 5-Pin CB Package



DESCRIPTION

The Allegro™ ACS758 family of current sensor ICs provides economical and precise solutions for AC or DC current sensing. Typical applications include motor control, load detection and management, power supply and DC-to-DC converter control, inverter control, and overcurrent fault detection.

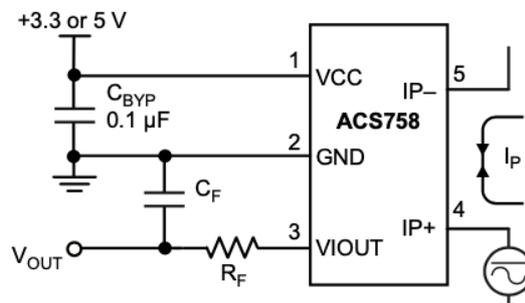
The device consists of a precision, low-offset linear Hall circuit with a copper conduction path located near the die. Applied current flowing through this copper conduction path generates a magnetic field which the Hall IC converts into a proportional voltage. Device accuracy is optimized through the close proximity of the magnetic signal to the Hall transducer. A precise, proportional output voltage is provided by the low-offset, chopper-stabilized BiCMOS Hall IC, which is programmed for accuracy at the factory.

High-level immunity to current conductor dV/dt and stray electric fields, offered by Allegro proprietary integrated shield technology, provides low output voltage ripple and low offset drift in high-side, high-voltage applications.

The output of the device has a positive slope ($>V_{CC}/2$) when an increasing current flows through the primary copper conduction path (from terminal 4 to terminal 5), which is the path used for current sampling. The internal resistance of this conductive path is 100 $\mu\Omega$ typical, providing low power loss.

The thickness of the copper conductor allows survival of the device at high overcurrent conditions. The terminals of the conductive path are electrically isolated from the signal leads

Continued on the next page...



Application 1: The ACS758 outputs an analog signal, V_{OUT} , that varies linearly with the uni- or bi-directional AC or DC primary sampled current, I_P , within the range specified. C_F is for optimal noise management, with values that depend on the application.

Typical Application

ACS758xCB

Thermally Enhanced, Fully Integrated, Hall-Effect-Based Linear Current Sensor IC with 100 $\mu\Omega$ Current Conductor

FEATURES AND BENEFITS (CONTINUED)

- 3.0 to 5.5 V, single supply operation
- 120 kHz typical bandwidth
- 3 μs output rise time in response to step input current
- Output voltage proportional to AC or DC currents
- Factory-trimmed for accuracy
- Extremely stable output offset voltage
- Nearly zero magnetic hysteresis

DESCRIPTION (CONTINUED)

(pins 1 through 3). This allows the ACS758 family of sensor ICs to be used in applications requiring electrical isolation without the use of opto-isolators or other costly isolation techniques.

The device is fully calibrated prior to shipment from the factory. The ACS758 family is lead (Pb) free. All leads are plated with 100% matte tin, and there is no Pb inside the package. The heavy gauge leadframe is made of oxygen-free copper.



Selection Guide

Part Number [1]	Package		Primary Sampled Current, I_P (A)	Sensitivity Sens (Typ.) (mV/A)	Current Directionality	T_{OP} ($^{\circ}\text{C}$)	Packing [2]
	Terminals	Signal Pins					
ACS758LCB-050B-PFF-T	Formed	Formed	± 50	40	Bidirectional	-40 to 150	34 pieces per tube
ACS758LCB-050U-PFF-T	Formed	Formed	50	60	Unidirectional		
ACS758LCB-100B-PFF-T	Formed	Formed	± 100	20	Bidirectional		
ACS758LCB-100B-PSF-T	Straight	Formed	± 100	20	Bidirectional		
ACS758LCB-100U-PFF-T	Formed	Formed	100	40	Unidirectional		
ACS758KCB-150B-PFF-T	Formed	Formed	± 150	13.3	Bidirectional	-40 to 125	
ACS758KCB-150U-PSF-T	Straight	Formed	150	26.7	Unidirectional		
ACS758KCB-150B-PSS-T	Straight	Straight	± 150	13.3	Bidirectional		
ACS758KCB-150U-PFF-T	Formed	Formed	150	26.7	Unidirectional		
ACS758ECB-200B-PFF-T	Formed	Formed	± 200	10	Bidirectional	-40 to 85	
ACS758ECB-200B-PSF-T	Straight	Formed	± 200	10	Bidirectional		
ACS758ECB-200U-PSF-T	Straight	Formed	200	20	Unidirectional		
ACS758ECB-200B-PSS-T	Straight	Straight	± 200	10	Bidirectional		
ACS758ECB-200U-PFF-T	Formed	Formed	200	20	Unidirectional		

¹ Additional leadform options available for qualified volumes.

² Contact Allegro for additional packing options.

ACS758xCB

Thermally Enhanced, Fully Integrated, Hall-Effect-Based Linear Current Sensor IC with 100 $\mu\Omega$ Current Conductor

SPECIFICATIONS

ABSOLUTE MAXIMUM RATINGS

Characteristic	Symbol	Notes	Rating	Units
Forward Supply Voltage	V_{CC}		8	V
Reverse Supply Voltage	V_{RCC}		-0.5	V
Forward Output Voltage	V_{IOUT}		28	V
Reverse Output Voltage	V_{RIOUT}		-0.5	V
Output Source Current	$I_{OUT(SOURCE)}$	V _{IOUT} to GND	3	mA
Output Sink Current	$I_{OUT(SINK)}$	V _{CC} to V _{IOUT}	1	mA
Nominal Operating Ambient Temperature	T_{OP}	Range E	-40 to 85	°C
		Range K	-40 to 125	°C
		Range L	-40 to 150	°C
Maximum Junction	$T_{J(max)}$		165	°C
Storage Temperature	T_{stg}		-65 to 165	°C

ISOLATION CHARACTERISTICS

Characteristic	Symbol	Notes	Rating	Unit
Dielectric Strength Test Voltage [1]	V_{ISO}	Agency type-tested for 60 seconds per UL standard 60950-1, 2nd Edition	4800	VAC
Working Voltage for Basic Isolation	V_{WFSI}	For basic (single) isolation per UL standard 60950-1, 2nd Edition	990	VDC or V_{pk}
			700	V_{rms}
Working Voltage for Reinforced Isolation	V_{WFRI}	For reinforced (double) isolation per UL standard 60950-1, 2nd Edition	636	VDC or V_{pk}
			450	V_{rms}

¹ Allegro does not conduct 60-second testing. It is done only during the UL certification process.



MICROCHIP

PIC24FJ256DA210 FAMILY

64/100-Pin, 16-Bit Flash Microcontrollers with Graphics Controller and USB On-The-Go (OTG)

Graphics Controller Features:

- Three Graphics Hardware Accelerators to Facilitate Rendering of Block Copying, Text and Unpacking of Compressed Data
- Color Look-up Table (CLUT) with Maximum of 256 Entries
- 1/2/4/8/16 bits-per-pixel (bpp) Color Depth Set at Run Time
- Display Resolution Programmable According to Frame Buffer:
 - Supports direct access to external memory on devices with EPMP
 - Resolution supported is up to 480x272 @ 60 Hz, 16 bpp; 640x480 @ 30 Hz, 16 bpp or 640x480 @ 60 Hz, 8 bpp
- Supports Various Display Interfaces:
 - 4/8/16-bit Monochrome STN
 - 4/8/16-bit Color STN
 - 9/12/18/24-bit Color TFT (18 and 24-bit displays are connected as 16-bit, 5-6-5 RGB color format)

Universal Serial Bus Features:

- USB v2.0 On-The-Go (OTG) Compliant
- Dual Role Capable – Can act as either Host or Peripheral
- Low-Speed (1.5 Mbps) and Full-Speed (12 Mbps) USB Operation in Host mode
- Full-Speed USB Operation in Device mode
- High-Precision PLL for USB
- Supports up to 32 Endpoints (16 bidirectional):
 - USB module can use the internal RAM location from 0x800 to 0xFFFF as USB endpoint buffers
- On-Chip USB Transceiver with Interface for Off-Chip Transceiver
- Supports Control, Interrupt, Isochronous and Bulk Transfers
- On-Chip Pull-up and Pull-Down Resistors

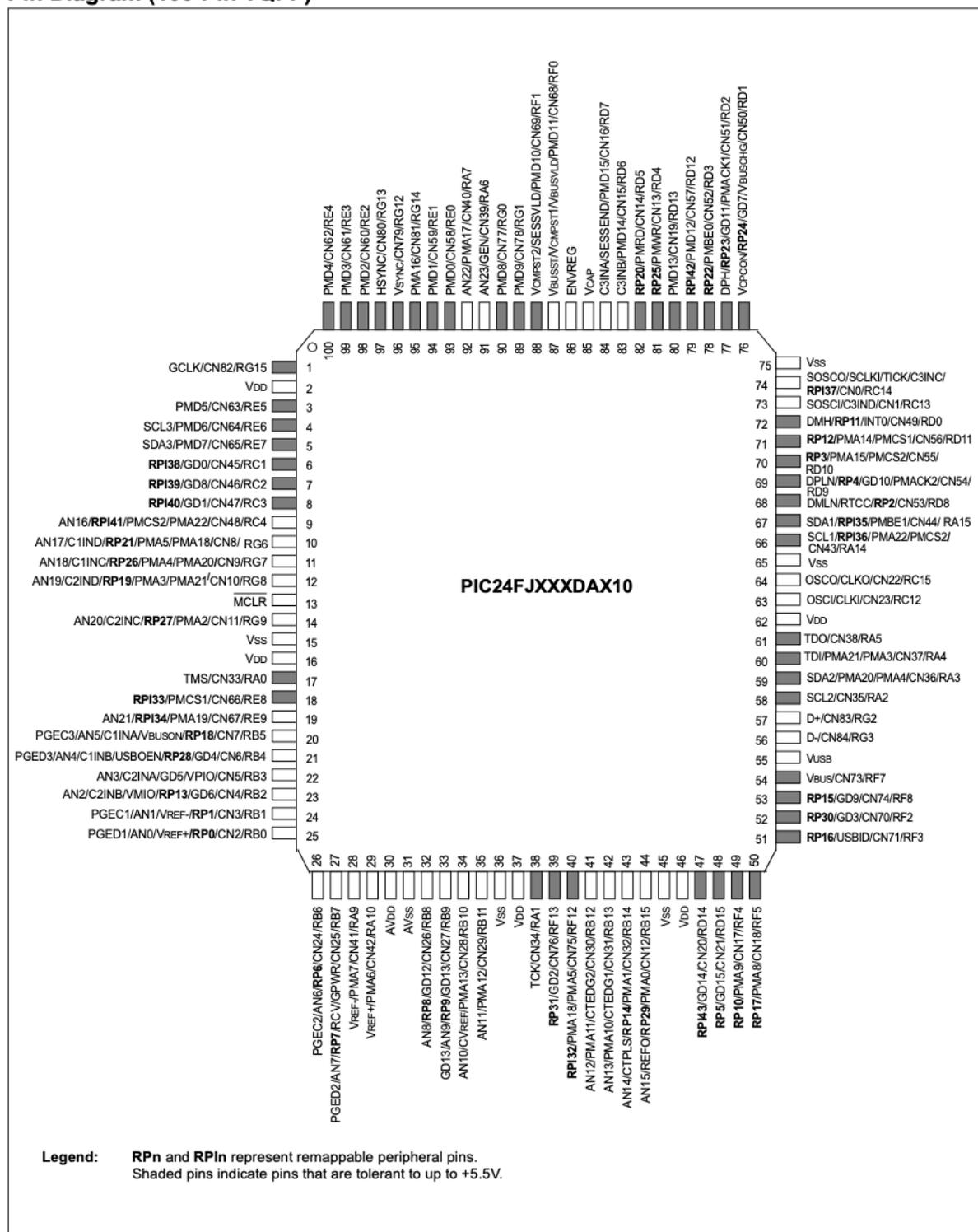
Peripheral Features:

- Enhanced Parallel Master Port/Parallel Slave Port (EPMP/PSP), 100-pin devices only:
 - Direct access from CPU with an Extended Data Space (EDS) interface
 - 4, 8 and 16-bit wide data bus
 - Up to 23 programmable address lines
 - Up to 2 chip select lines
 - Up to 2 Acknowledgement lines (one per chip select)
 - Programmable address/data multiplexing
 - Programmable address and data Wait states
 - Programmable polarity on control signals
- Peripheral Pin Select:
 - Up to 44 available pins (100-pin devices)
- Three 3-Wire/4-Wire SPI modules (supports 4 Frame modes)
- Three I²C™ modules Supporting Multi-Master/Slave modes and 7-Bit/10-Bit Addressing
- Four UART modules:
 - Supports RS-485, RS-232, LIN/J2602 protocols and IrDA®
- Five 16-Bit Timers/Counters with Programmable Prescaler
- Nine 16-Bit Capture Inputs, each with a Dedicated Time Base
- Nine 16-Bit Compare/PWM Outputs, each with a Dedicated Time Base
- Hardware Real-Time Clock and Calendar (RTCC)
- Enhanced Programmable Cyclic Redundancy Check (CRC) Generator
- Up to 5 External Interrupt Sources

PIC24FJ Device	Pins	Program Memory (bytes)	SRAM (bytes)	Remappable Peripherals					I ² C™	10-Bit A/D (ch)	Comparators	CTMU	EPMP/PSP	RTCC	Graphics Controller	USB OTG
				Remappable Pins	16-Bit Timers	IC/OC PWM	UART w/IrDA®	SPI								
PIC24FJ128DA106	64	128K	24K	29	5	9/9	4	3	3	16	3	Y	N	Y	Y	Y
PIC24FJ256DA106	64	256K	24K	29	5	9/9	4	3	3	16	3	Y	N	Y	Y	Y
PIC24FJ128DA110	100/121	128K	24K	44	5	9/9	4	3	3	24	3	Y	Y	Y	Y	Y
PIC24FJ256DA110	100/121	256K	24K	44	5	9/9	4	3	3	24	3	Y	Y	Y	Y	Y
PIC24FJ128DA206	64	128K	96K	29	5	9/9	4	3	3	16	3	Y	N	Y	Y	Y
PIC24FJ256DA206	64	256K	96K	29	5	9/9	4	3	3	16	3	Y	N	Y	Y	Y
PIC24FJ128DA210	100/121	128K	96K	44	5	9/9	4	3	3	24	3	Y	Y	Y	Y	Y
PIC24FJ256DA210	100/121	256K	96K	44	5	9/9	4	3	3	24	3	Y	Y	Y	Y	Y

PIC24FJ256DA210 FAMILY

Pin Diagram (100-Pin TQFP)



PIC24FJ256DA210 FAMILY

TABLE 2: COMPLETE PIN FUNCTION DESCRIPTIONS FOR 100-PIN DEVICES

Pin	Function	Pin	Function
1	GCLK/CN82/RG15	41	AN12/PMA11/CTEDG2/CN30/RB12
2	Vdd	42	AN13/PMA10/CTEDG1/CN31/RB13
3	PMD5/CN63/RE5	43	AN14/CTPLS/RP14/PMA1/CN32/RB14
4	SCL3/PMD6/CN64/RE6	44	AN15/REFO/RP29/PMA0/CN12/RB15
5	SDA3/PMD7/CN65/RE7	45	Vss
6	RPI38/GD0/CN45/RC1	46	VDD
7	RPI39/GD8/CN46/RC2	47	RPI43/GD14/CN20/RD14
8	RPI40/GD1/CN47/RC3	48	RP5/GD15/CN21/RD15
9	AN16/RPI41/PMCS2/PMA22 ⁽²⁾ /CN48/RC4	49	RP10/PMA9/CN17/RF4
10	AN17/C1IND/RP21/PMA5/PMA18 ⁽²⁾ /CN8/RG6	50	RP17/PMA8/CN18/RF5
11	AN18/C1INC/RP26/PMA4/PMA20 ⁽²⁾ /CN9/RG7	51	RP16/USBID/CN71/RF3
12	AN19/C2IND/RP19/PMA3/PMA21 ⁽²⁾ /CN10/RG8	52	RP30/GD3/CN70/RF2
13	MCLR	53	RP15/GD9/CN74/RF8
14	AN20/C2INC/RP27/PMA2/CN11/RG9	54	VBus/CN73/RF7
15	Vss	55	VUSB
16	Vdd	56	D-/CN84/RG3
17	TMS/CN33/RA0	57	D+/CN83/RG2
18	RPI33/PMCS1/CN66/RE8	58	SCL2/CN35/RA2
19	AN21/RPI34/PMA19/CN67/RE9	59	SDA2/PMA20/PMA4 ⁽²⁾ /CN36/RA3
20	PGEC3/AN5/C1INA/VBusON/RP18/CN7/RB5	60	TDI/PMA21/PMA3 ⁽²⁾ /CN37/RA4
21	PGED3/AN4/C1INB/USBOEN/RP28/GD4/CN6/RB4	61	TDO/CN38/RA5
22	AN3/C2INA/GD5/VPIO/CN5/RB3	62	Vdd
23	AN2/C2INB/VMIO/RP13/GD6/CN4/RB2	63	OSCI/CLKI/CN23/RC12
24	PGEC1/AN1/VREF ⁽¹⁾ /RP1/CN3/RB1	64	OSCO/CLKO/CN22/RC15
25	PGED1/AN0/VREF ⁽¹⁾ /RP0/CN2/RB0	65	Vss
26	PGEC2/AN6/RP6/CN24/RB6	66	SCL1/RPI36/PMA22/PMCS2 ⁽²⁾ /CN43/RA14
27	PGED2/AN7/RP7/RCV/GPWR/CN25/RB7	67	SDA1/RPI35/PMBE1/CN44/RA15
28	VREF-/PMA7/CN41/RA9	68	DMLN/RTCC/RP2/CN53/RD8
29	VREF+/PMA6/CN42/RA10	69	DPLN/RP4/GD10/PMACK2/CN54/RD9
30	AVDD	70	RP3/PMA15/PMCS2 ⁽³⁾ /CN55/RD10
31	AVss	71	RP12/PMA14/PMCS1 ⁽³⁾ /CN56/RD11
32	AN8/RP8/GD12/CN26/RB8	72	DMH/RP11/INT0/CN49/RD0
33	AN9/RP9/GD13/CN27/RB9	73	SOSCI/C3IND/CN1/RC13
34	AN10/CVREF/PMA13/CN28/RB10	74	SOSCO/SCLKI/T1CK/C3INC/RPI37/CN0/RC14
35	AN11/PMA12/CN29/RB11	75	Vss
36	Vss	76	VcPCON/RP24/GD7/VBusCHG/CN50/RD1
37	Vdd	77	DPH/RP23/GD11/PMACK1/CN51/RD2
38	TCK/CN34/RA1	78	RP22/PMBE0/CN52/RD3
39	RP31/GD2/CN76/RF13	79	RPI42/PMD12/CN57/RD12
40	RPI32/PMA18/PMA5 ⁽²⁾ /CN75/RF12	80	PMD13/CN19/RD13

Legend: RPn and RPin represent remappable pins for Peripheral Pin Select (PPS) functions.

Note 1: Alternate pin assignments for VREF+ and VREF- when the ALTVREF Configuration bit is programmed.

Note 2: Alternate pin assignments for EPMP when the ALTPMP Configuration bit is programmed.

Note 3: Pin assignment for PMCSx when CSF<1:0> is not equal to '00'.

PIC24FJ256DA210 FAMILY

TABLE 2: COMPLETE PIN FUNCTION DESCRIPTIONS FOR 100-PIN DEVICES

Pin	Function	Pin	Function
81	RP25/PMWR/CN13/RD4	91	AN23/GEN/CN39/RA6
82	RP20/PMRD/CN14/RD5	92	AN22/PMA17/CN40/RA7
83	C3INB/PMD14/CN15/RD6	93	PMD0/CN58/RE0
84	C3INA/SESEND/PMD15/CN16/RD7	94	PMD1/CN59/RE1
85	VCAP	95	PMA16/CN81/RG14
86	ENVREG	96	VSYNC/CN79/RG12
87	VBUSST/VCMPST1/VBUSVLD/PMD11/CN68/RF0	97	HSYNC/CN80/RG13
88	VCMPST2/SESSVLD/PMD10/CN69/RF1	98	PMD2/CN60/RE2
89	PMD9/CN78/RG1	99	PMD3/CN61/RE3
90	PMD8/CN77/RG0	100	PMD4/CN62/RE4

- Legend:** RPN and RPin represent remappable pins for Peripheral Pin Select (PPS) functions.
- Note**
- 1: Alternate pin assignments for VREF+ and VREF- when the ALTVREF Configuration bit is programmed.
 - 2: Alternate pin assignments for EPMP when the ALTPMP Configuration bit is programmed.
 - 3: Pin assignment for PMCSx when CSF<1:0> is not equal to '00'.

PIC24FJ256DA210 FAMILY

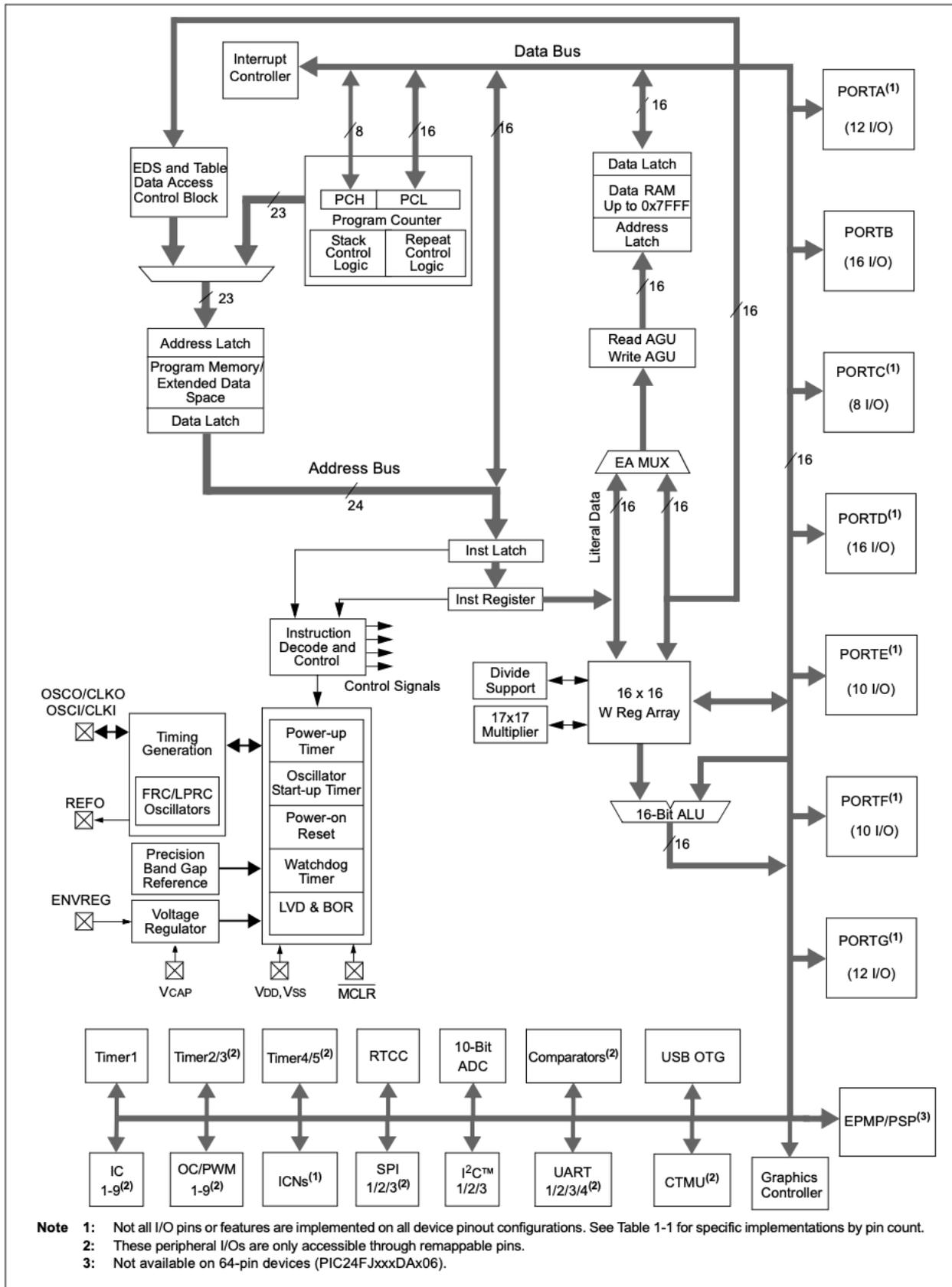
TABLE 1-2: DEVICE FEATURES FOR THE PIC24FJ256DA210 FAMILY: 100-PIN DEVICES

Features	PIC24FJ128DA110	PIC24FJ256DA110	PIC24FJ128DA210	PIC24FJ256DA210
Operating Frequency	DC – 32 MHz			
Program Memory (bytes)	128K	256K	128K	256K
Program Memory (instructions)	44,032	87,552	44,032	87,552
Data Memory (bytes)	24K		96K	
Interrupt Sources (soft vectors/NMI traps)	66 (62/4)			
I/O Ports	Ports A, B, C, D, E, F, G			
Total I/O Pins	84			
Remappable Pins	44 (32 I/O, 12 input only)			
Timers:				
Total Number (16-bit)	5 ⁽¹⁾			
32-Bit (from paired 16-bit timers)	2			
Input Capture Channels	9 ⁽¹⁾			
Output Compare/PWM Channels	9 ⁽¹⁾			
Input Change Notification Interrupt	84			
Serial Communications:				
UART	4 ⁽¹⁾			
SPI (3-wire/4-wire)	3 ⁽¹⁾			
I ² C™	3			
Parallel Communications (EPMP/PSP)	Yes			
JTAG Boundary Scan	Yes			
10-Bit Analog-to-Digital Converter (ADC) Module (input channels)	24			
Analog Comparators	3			
CTMU Interface	Yes			
USB OTG	Yes			
Graphics Controller	Yes			
Resets (and delays)	POR, BOR, RESET Instruction, MCLR, WDT; Illegal Opcode, REPEAT Instruction, Hardware Traps, Configuration Word Mismatch (OST, PLL Lock)			
Instruction Set	76 Base Instructions, Multiple Addressing Mode Variations			
Packages	100-Pin TQFP and 121-Pin BGA			

Note 1: Peripherals are accessible through remappable pins.

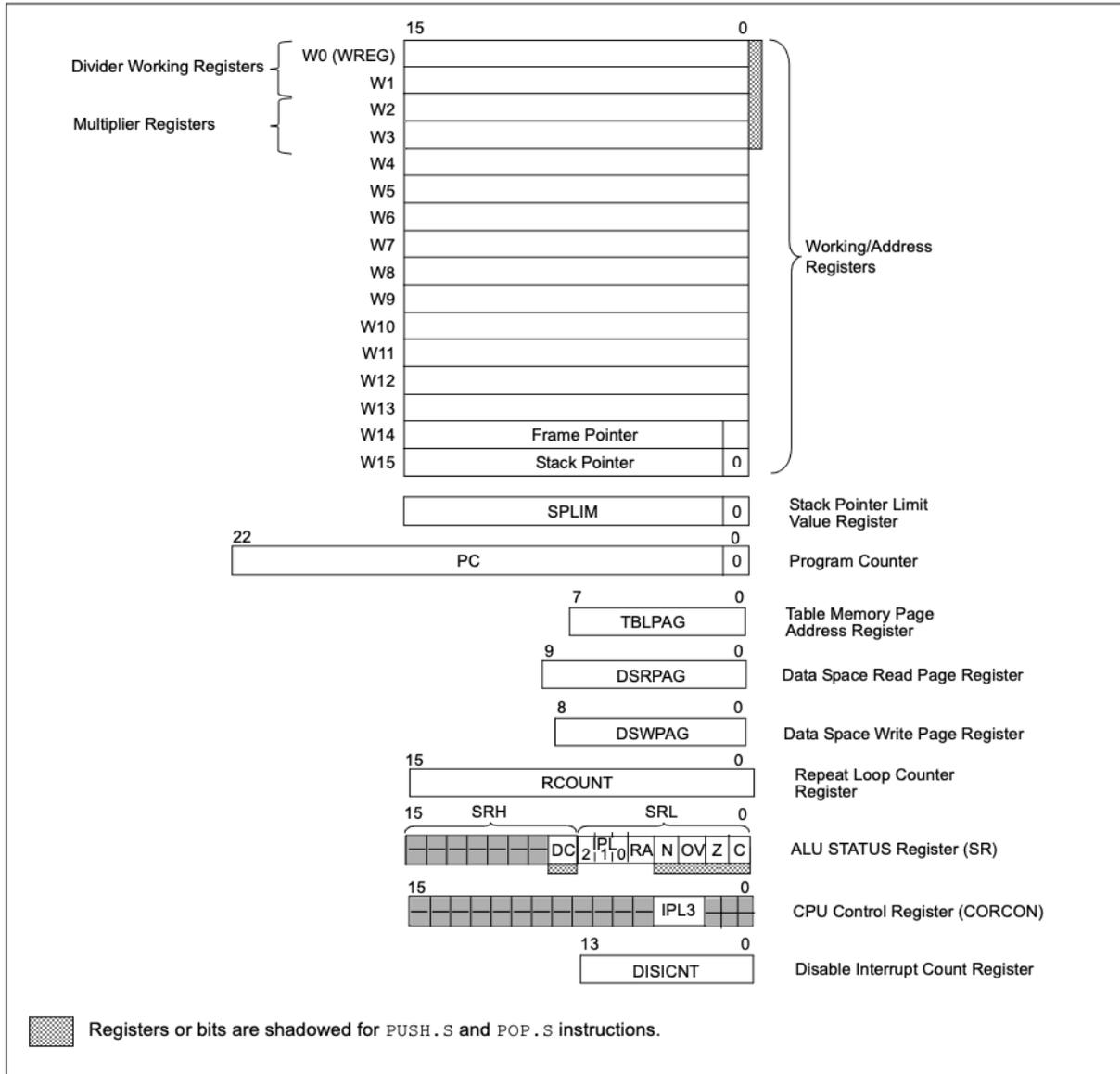
PIC24FJ256DA210 FAMILY

FIGURE 1-1: PIC24FJ256DA210 FAMILY GENERAL BLOCK DIAGRAM



PIC24FJ256DA210 FAMILY

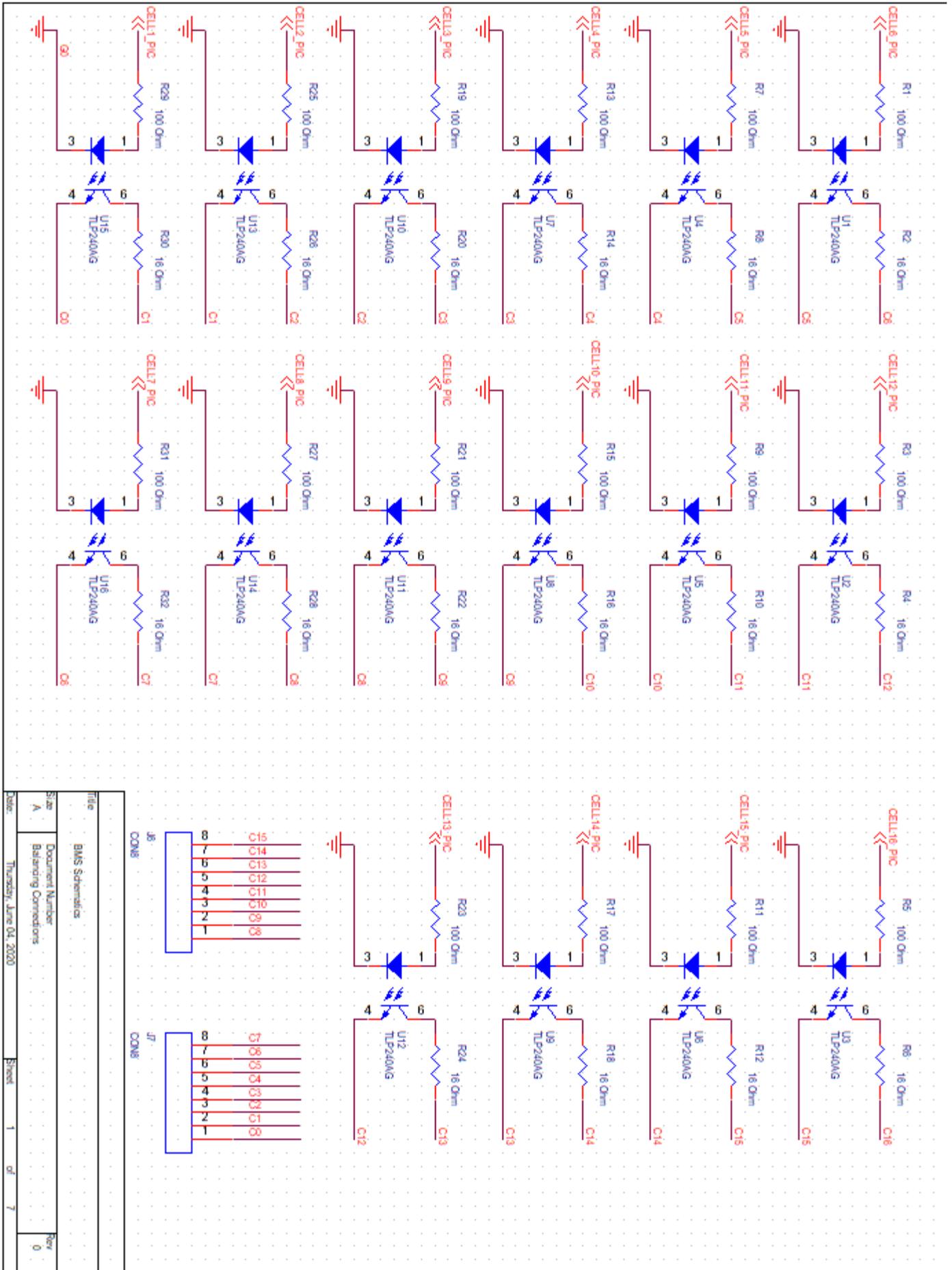
FIGURE 3-2: PROGRAMMER'S MODEL



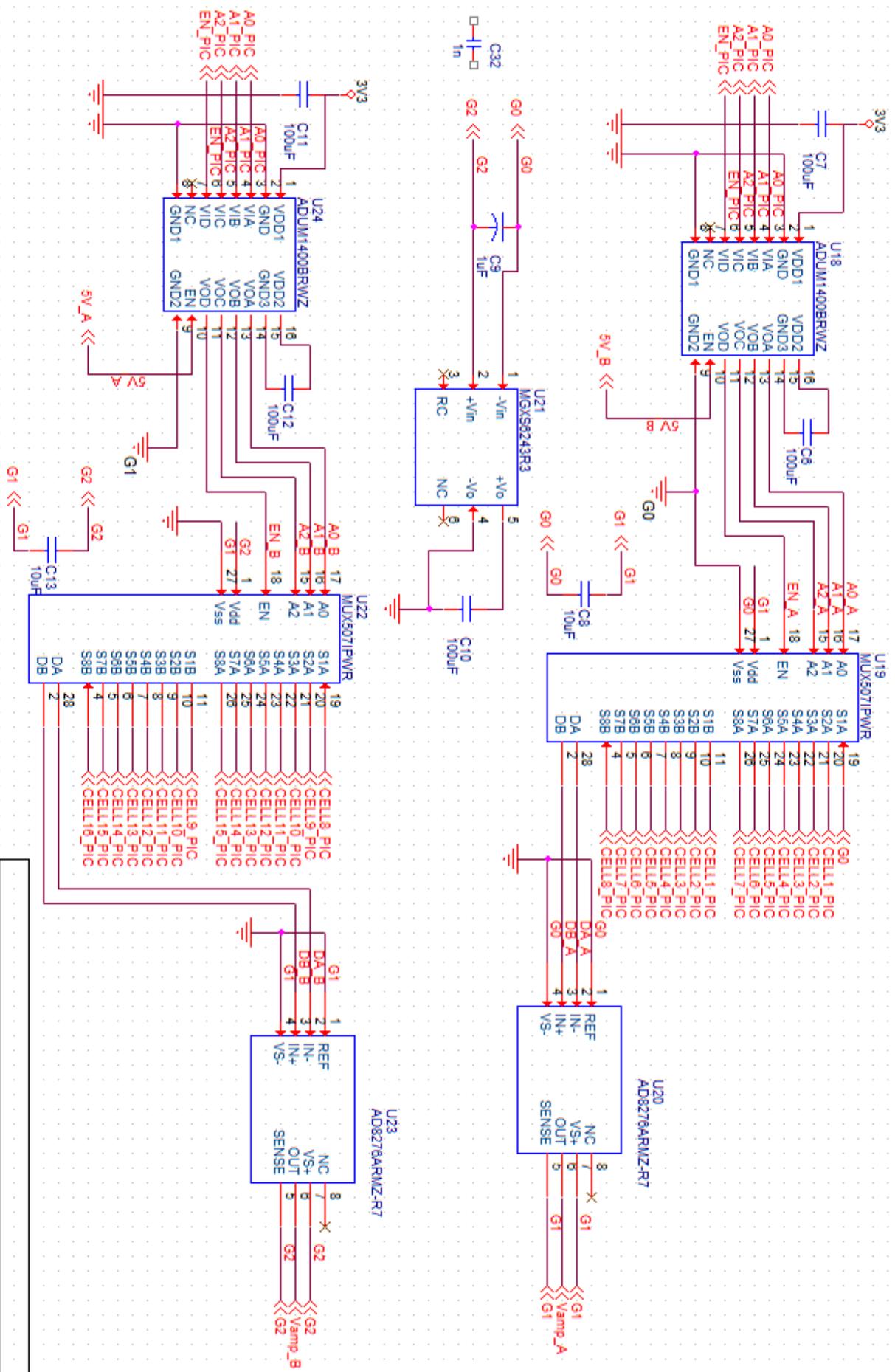
ANEXO 2: ESQUEMAS ELECTRÓNICOS



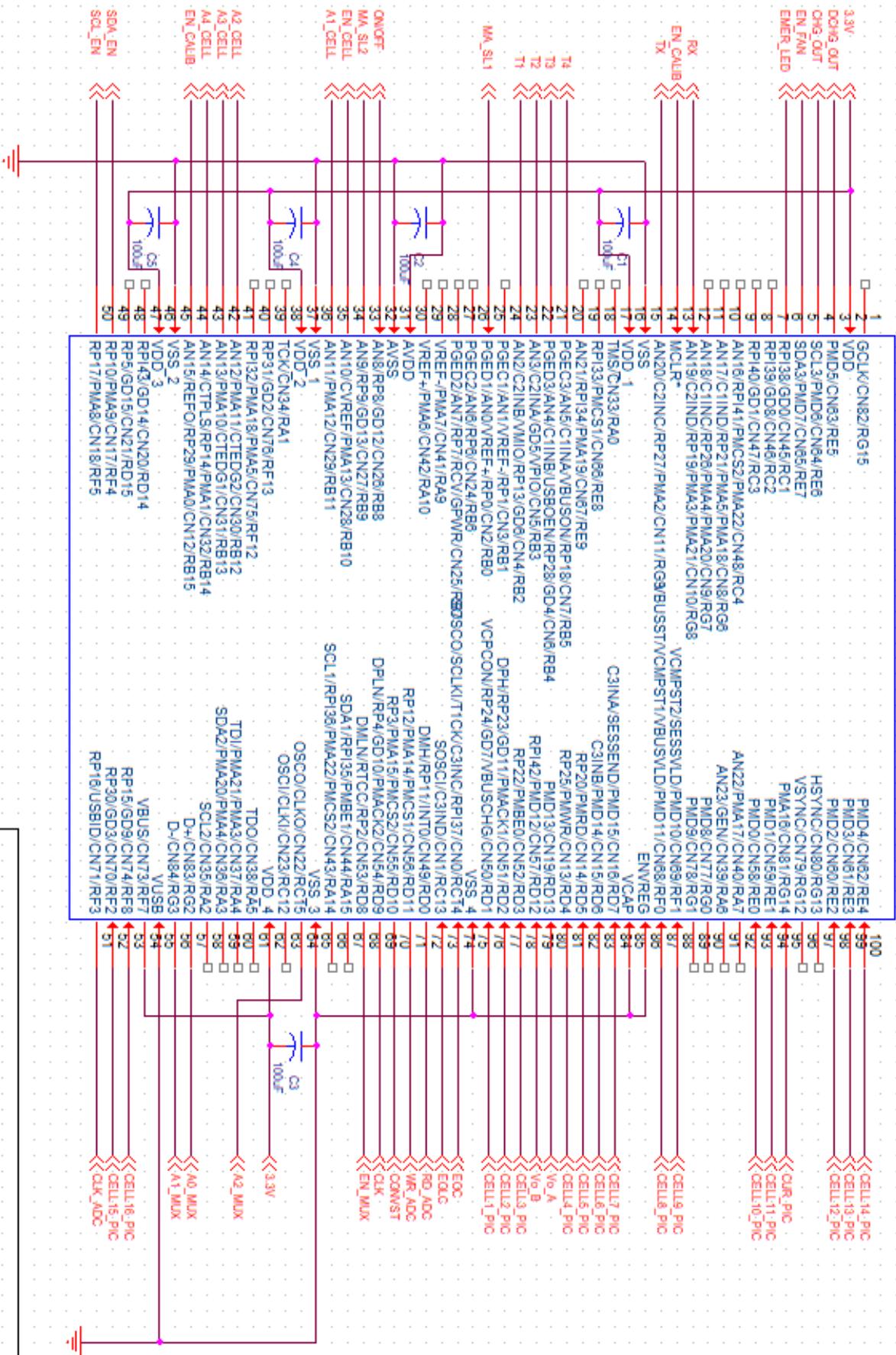
UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



File	BMS Schematics	
Size	Document Number	Rev
A	Balancing Connectors	0
Date	Thursday, June 04, 2020	Sheet 1 of 7

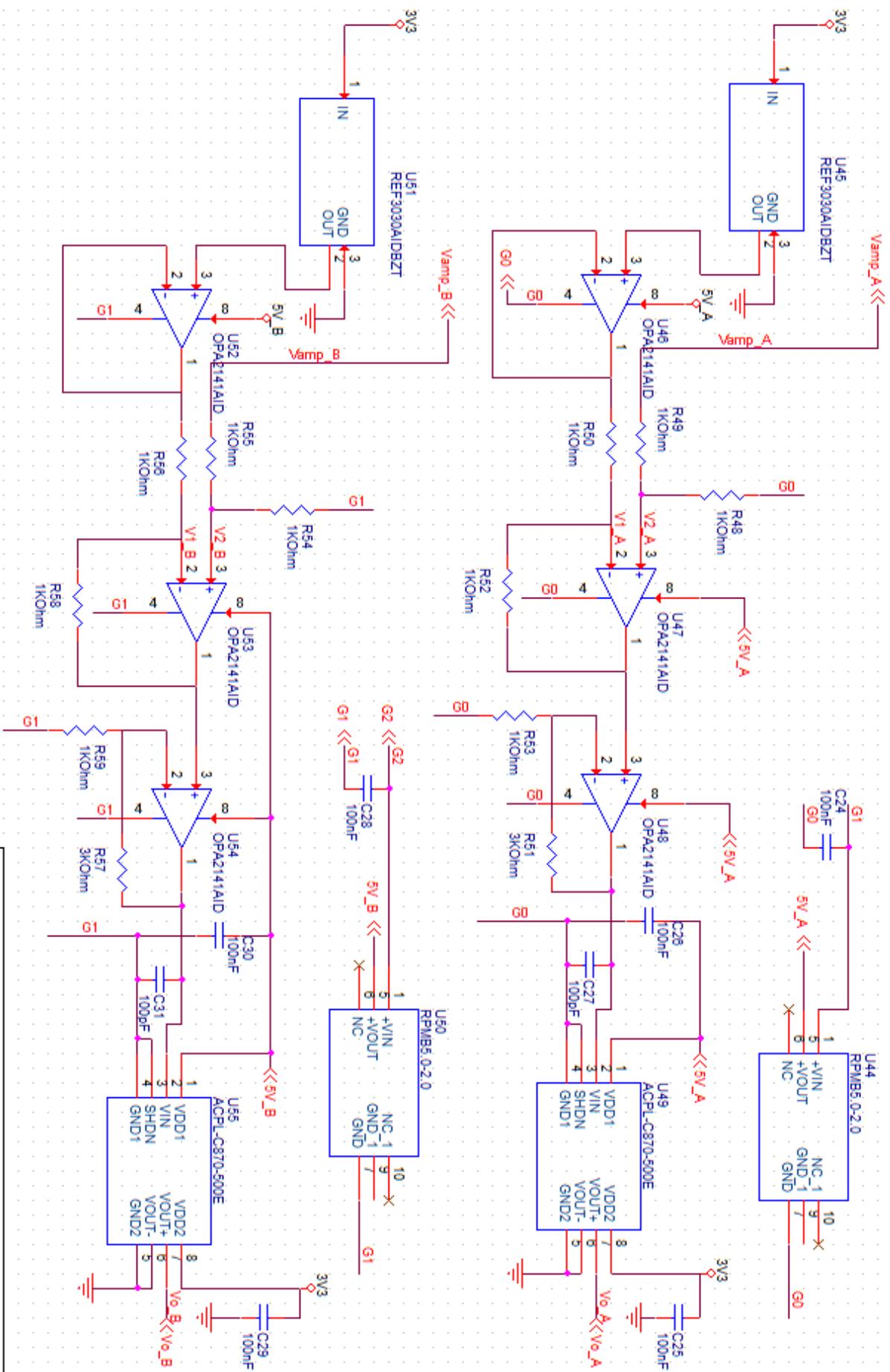


Title		BMS Schematics
Size	Document Number	Analogue Multiplexers
A	Rev	0
Date:	Tuesday, August 18, 2020	Sheet 3 of 7

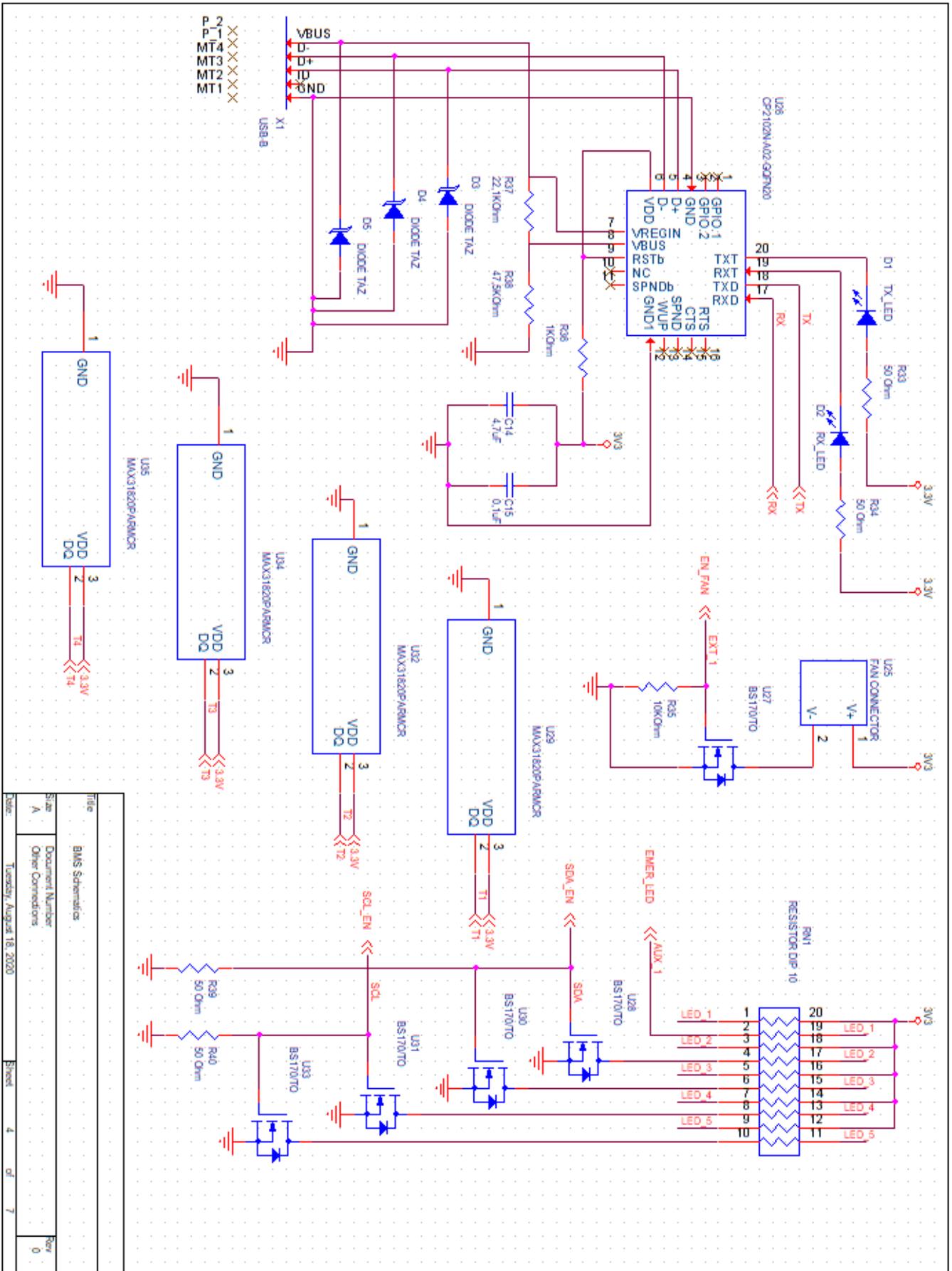


U17
PIC24F2560A210T-PT

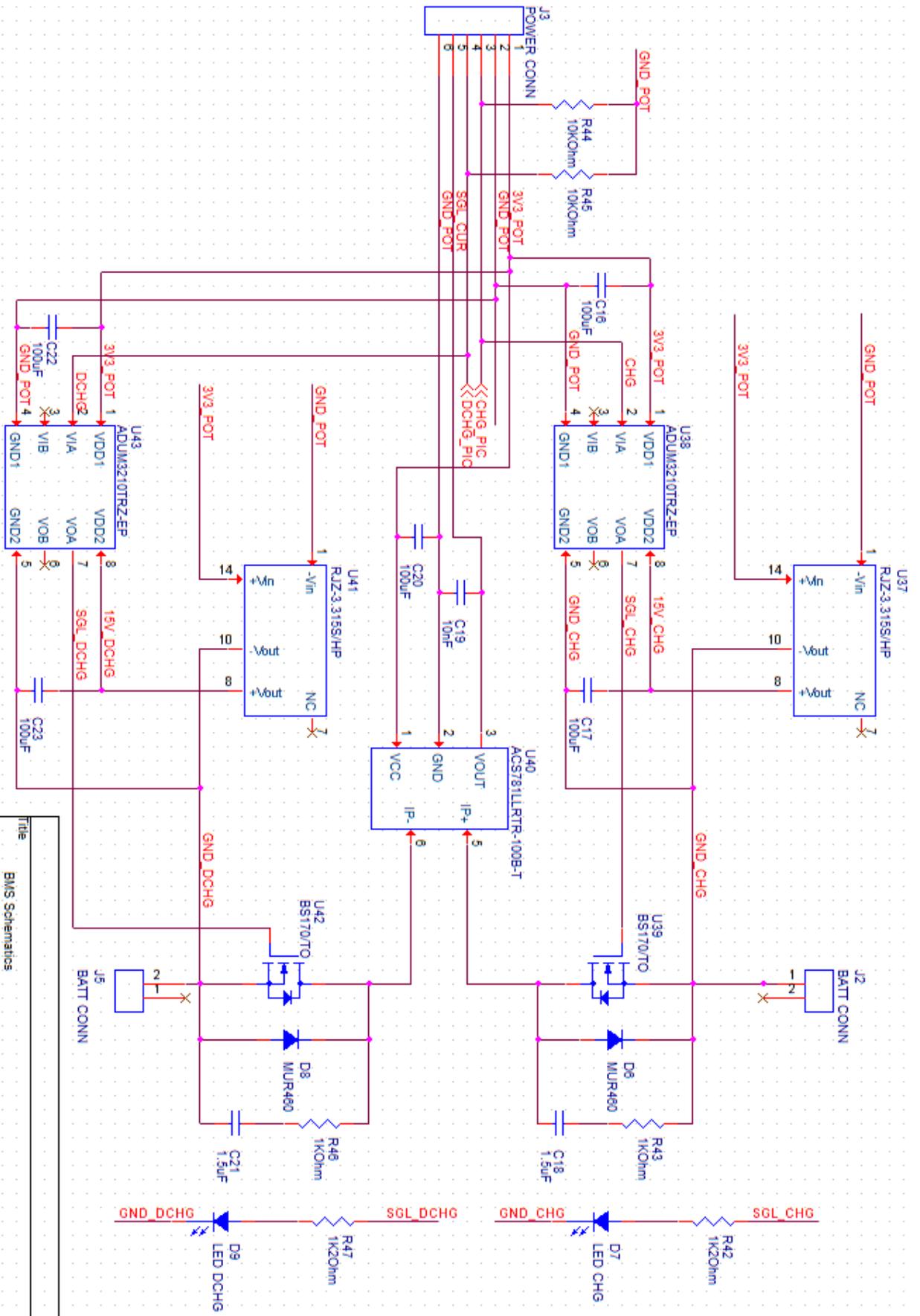
Title		BMS Schematics	
Document Number		PIC Microcontroller	
Size	A	Sheet	2 of 7
Date:	Tuesday, August 18, 2020	Rev	0



Title	
BMS Schematics	
Size	Document Number
A	Balancing Circuitry
Rev	Date
<Rev	Tuesday, August 18, 2020
	Sheet 7 of 7



File	BMS Schematics
Size	Document Number
A	Other Connections
Date	Tuesday, August 18, 2020
Sheet	4 of 7
Rev	0



Title		BMS Schematics
Size		Document Number
Rev		0
Date		Tuesday, August 13, 2020
Sheet		3 of 7

ANEXO 3:

PROGRAMACIÓN

DEL MICRO PIC



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

```

/*
 * File: main.c
 * Author: Ahullonet
 *
 * Created on July 19, 2020, 7:12 PM
 */

#define FOSC (32000000)
#define FCY (FOSC/2)

//-----INCLUDES-----

#include <libpic30.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <xc.h>
#include "auxiliar.h"
#include "initialize.h"
#include "readsymux.h"
#include "ADC.h"
#include "I2C.h"
#include "structs.h"
#include "thermos.h"
#include "balancing.h"

// Config 1
#pragma config JTAGEN = OFF
// JTAG Port Enable (JTAG port is disabled)
#pragma config GCP = ON
// General Segment Code Protect (Code protection is enabled)
#pragma config GWRP = OFF
// General Segment Write Protect (Writes to program memory are allowed)

```

```

#pragma config ICS = PGx2
// Emulator Pin Placement Select bits (Emulator functions are shared with PGEC2)
#pragma config FWDTEN = OFF
// Watchdog Timer) Watchdog timer is disabled)
#pragma config WINDIS = OFF
// Windowed WDT (Standard Watchdog Timer Enabled, (Windowed mode is
disabled))
#pragma config FWPSA = PR128
// WDT Prescaler (Prescaler ratio of 1:128)
#pragma config WDTPS = PS32768
// Watchdog Timer Postscaler (1:32,768)

// Config 2
#pragma config IESO = ON
#pragma config PLLDIV = DIV2
#pragma config PLL96MHZ = ON
#pragma config FNOSC = FRCPLL
#pragma config FCKSM = CSDCMD
#pragma config OSCIOFNC = ON
#pragma config IOL1WAY = ON
#pragma config POSCMOD = NONE

//Config 3
#pragma config WPEND = WPENDMEM
#pragma config WPCFG = WPCFGDIS
#pragma config WPDIS = WPDIS
#pragma config ALTPMP = ALPMPDIS
#pragma config WUTSEL = LEG
#pragma config SOSSEL = EC
#pragma config WPFPP = WPFPP255

```

```
Tfeedback Response(char RX_data);
```

```
int main(int argc, char** argv) {
```

```
    int fault, init;
```

```
    int SW_ON;
```

```
    init = 1;
```

```
    printf("El programa ha empezado porque Init es %i\n", init);
```

```
    Initialize();
```

```
    printf("El programa ha inicializado!\n");
```

```
    SW_ON = 1;
```

```
    printf("SW_ON = %i\n", SW_ON);
```

```
    while (1) {
```

```
        if ((SW_ON == 0) && (get_MA_value() == 1))    // Equipo maestro apagado
```

```
        {
```

```
            printf("El equipo esta apagado!\n");
```

```
            all_disabled ();           // Desactivar el BMS
```

```
            printf("Desactivacion del BMS!\n");
```

```
            increase_counter_T_value();
```

```
            if (get_counter_T_value() == 9999999)
```

```
            {
```

```
                state_verify();
```

```
                set_counter_T_value(0);
```

```
                printf("Reseteo del contador\n");
```

```
            }
```

```
            set_DCHG_CHG_value(3);
```

```
            printf("Se asigna el modo espera, WAITING MODE\n");
```

```
            // Aunque esté en el estado 3 no se permite la carga ni la descarga,
```

```
            // pero se pone para que se active al encender
```

```

}
if ((SW_ON == 1) && (get_MA_value()==1)) // Equipo maestro encendido
{
    printf("Equipo maestro encendido\n");
    init_Thermos (); // Activación de la lectura de los termómetros
    printf("Sensores de temperatura inicializados\n");
    read_data (); // Lectura del ADC
    printf("Datos del ADC leídos\n");
    fault = limit_verify (); // Verificación de los márgenes de funcionamiento
    fault = 0;
    if ((fault==1) || (fault==4) || (fault==5)) // Comprobación de fallos de T°, I o
    celdas incorrectas
    {
        printf("Se ha detectado un fallo grave de temperatura, corriente o conexión
de celdas\n");
        set_DCHG_CHG_value(0); // No se permite ni carga ni descarga
        calib_disable ();
    }
    set_DCHG_CHG_value(3);
    set_Current_value(-10416);
    fault = 0;
if ((fault!=1) && (fault!=4)) // El equipo encendido y sin fallos graves
{
    printf("Entramos en la elección de modos");
    if ((get_Current_value() > POSITIVE_SENS_A) &&
((get_DCHG_CHG_value()==3) || (get_DCHG_CHG_value()==1)))
    {
        DCHG_mode (); // Modo descarga
        printf("MODO DESCARGA\n");
    }
    if ((get_Current_value() < NEGATIVE_SENS_A) &&
((get_DCHG_CHG_value()==3) || (get_DCHG_CHG_value()==2)))
    {
        CHG_mode (); // Modo carga
    }
}

```

```

    printf("MODO CARGA\n");
}
if ((get_Current_value() < NEGATIVE_SENS_B) && (get_Current_value() <
POSITIVE_SENS_B))
{
    WAITING_mode (); // Modo espera
    printf("MODO ESPERA\n");
}
}
USB_comm ();          // Revisión de comunicaciones USB
read_temp_Thermos (); // Lectura de los termómetros
MOSFET_enable ();    // Activar los MOSFET correspondientes
}
}
return (EXIT_SUCCESS);
}

```

```

////////////////////////////////////
// Rutina de desactivación del BMS
////////////////////////////////////

```

```

void all_disabled (void)
{
    DCHG = 0;          // Se deshabilita la descarga
    CHG = 0;          // Se deshabilita la carga
    LED_stripe (0);   // Se deshabilita la barra de LED's
    LATDbits.LATD8 = 0; // Desactivación de los multiplexores
    calib_disable (); // Desactivación del equilibrado de las celdas
    set_CHARGED_value(0);
}

```

```

////////////////////////////////////

```

```

// Rutina de verificación de estados
////////////////////////////////////

void state_verify (void)
{
    int fault;

    LATDbits.LATD8 = 1;    // Se habilitan los multiplexores
    set_DCHG_CHG_value(0);    // No se permite nada para que calibre el sensor
    read_data ();    // Se leen los datos
    fault = limit_verify (); // Se verifican los límites de funcionamiento
    if ((fault==1) || (fault==4)) EMER_LED = 1; // Encender el led de emergencia
}

////////////////////////////////////
// Verificación de los límites de seguridad
// 0 = Sin error
// 1 = Error de temperatura
// 2 = Celda sobredescargada
// 3 = Celda sobrecargada
// 4 = Error de corriente
// 5 = Celdas incorrectas
////////////////////////////////////

int limit_verify (void)
{
    printf("Se procede a la verificacion de los limites de los sensores\n");
    int error = 0, emer = 0;
    int i;
    int cell_inf_fault = 0;
    int cell_sup_fault = 0;
    long int current_peak_time = 0;
    Tbattery battery_info;

```

```

printf ("Sensores de temperatura:\n");
set_Temper1_value(25.0);
if ((get_Temper1_value()<9999) && (get_Temper1_value()> T_FAN)) FAN = 1;
// Activar el ventilador si el sensor mide bien y pasa de 50°
else FAN = 0;          // Si no se cumple, se deshabilita
printf("Primer sensor (ambiente) conectado y midiendo %f\n",
get_Temper1_value());
printf("Por tanto, ventilador igual a 0\n");
set_Temper2_value(25.0000);
if ((get_Temper2_value()<9999) && (get_Temper2_value()>
battery_info.SOA.t_max)) error = 0;
printf("Error T2 = %i\n", error);
set_Temper3_value(25.0);
if ((get_Temper3_value()<9999) && (get_Temper3_value()>
battery_info.SOA.t_max)) error = 0;
printf("Error T3 = %i\n", error);
set_Temper4_value(25.0);
if ((get_Temper4_value()<9999) && (get_Temper4_value()>
battery_info.SOA.t_max)) error = 0;
printf("Error T4 = %i\n", error);

if (error == 0) // Si no hay fallo de temperatura, continua verificando
{
printf("No hay fallo de temperatura, se continua verificando. ErrorT = %i\n",
error);

for (i=0;i<(N_BLOCK_CELLS);i++)
{
// Se mantiene el orden de verificación de las celdas
// para que no pueda cargar con una celda rota
if ((get_voltage_cells_pose(i)<battery_info.SOA.max_dchg_tension) &&
(i<battery_info.n_used_blockA_cells))
{
printf("Alguna celda del bloque A esta por debajo del voltaje minimo\n");
}
}
}

```

```

error = 2;
set_DCHG_CHG_value(2);      // Solamente se permite la carga
set_CHARGED_value(4);      // Estado de la batería sin cargar
cell_inf_fault = 2;

printf("DCHG_CHG = %i. Solamente permite la carga\n",
get_DCHG_CHG_value());

printf("CHARGED = %i. Estado de la bateria sin cargar\n",
get_CHARGED_value());
}
if
((get_voltage_cells_pose(N_BLOCK_CELLS+i)<battery_info.SOA.max_dchg_tensio
n) && (i<battery_info.n_used_blockB_cells))
{
printf("Alguna celda del bloque B esta por debajo del voltaje minimo\n");
error = 2;
set_DCHG_CHG_value(2);      // Solamente se permite la carga
set_CHARGED_value(4);      // Estado de la batería sin cargar
cell_inf_fault = 2;

printf("DCHG_CHG = %i. Solamente permite la carga\n",
get_DCHG_CHG_value());

printf("CHARGED = %i. Estado de la bateria sin cargar\n",
get_CHARGED_value());
}
printf ("Voltaje celda %i = %i\n", i, get_voltage_cells_pose(i));
if ((get_voltage_cells_pose(i)>battery_info.SOA.sec_tension) &&
(i<battery_info.n_used_blockA_cells))
{
printf("Alguna de las celdas del bloque A esta por encima del voltaje
maximo\n");
error = 3;
set_DCHG_CHG_value(1);      // Solamente se permite la descarga
set_CHARGED_value(3);      // Hay una celda del bloque A sobrecargada
cell_sup_fault = 3;

printf("DCHG_CHG = %i. Solamente permite la descarga\n",
get_DCHG_CHG_value());

printf("CHARGED = %i. Bateria cargada pero con una celda
sobrecargada\n", get_CHARGED_value());
}

```

```

//    }

    printf("Voltaje celda %i = %li\n",N_BLOCK_CELLS+i,
get_voltage_cells_pose(N_BLOCK_CELLS+i));

    if
((get_voltage_cells_pose(N_BLOCK_CELLS+i)>battery_info.SOA.sec_tension) &&
(i<battery_info.n_used_blockB_cells))

    {

        printf("La celda %i del bloque B esta por encima del voltaje maximo\n", i);
        printf("Error = 3\n");
        error = 3;

        set_DCHG_CHG_value(1);        // Solamente se permite la descarga
        set_CHARGED_value(3);        // Hay una celda del bloque B sobrecargada

        cell_sup_fault = 3;

        printf("DCHG_CHG = %i. Solamente permite la descarga\n",
get_DCHG_CHG_value());

        printf("CHARGED = %li. Bateria cargada pero con una celda
sobrecargada\n", get_CHARGED_value());

    }

}

//    printf("Se apaga el MOSFET de carga\n");

error = cell_inf_fault + cell_sup_fault;
if (error>1) {
    emer = 1;

    printf("Como hay error de sobrevoltaje, SE HABILITA EL ESTADO DE
EMERGENCIA\n");
}

else {emer = 0;}

    printf("NO hay error de voltaje, se sigue verificando\n");
    printf("La corriente instantanea es = %li\n", get_Current_value());

    if ((get_Current_value())<battery_info.max_dchg_current) &&
(get_Current_value())>battery_info.max_chg_current)

    {

        printf("La corriente instantanea = %li\n", get_Current_value());

```

```

    printf("La corriente esta dentro de los limites\n");
    current_peak_time = 0; // Corriente menor que la descarga máxima nominal
}
else current_peak_time = 0;
    printf("La corriente instantanea = %li\n", get_Current_value());
    printf("La corriente esta dentro de los limites\n");
//    printf("La corriente de descarga es mayor que el valor maximo\n");
//    printf("Empieza el contador del tiempo de pico admisible\n");

    if ((get_Current_value()>battery_info.peak_dchg_current) ||
        (get_Current_value()<battery_info.peak_chg_current) || (current_peak_time =
T_MAX_PEAK))
    {
//        printf("Error de corriente por causa del valor de carga, descarga, o tiempo
de pico\n");
        error = 0;
    }
}
set_Total_V_value(58096.000);
if (get_Total_V_value()<get_discharged_battery_value())
{
    emer = 1;        // Si la tensión total baja de este valor, el BMS corta
    set_DCHG_CHG_value(2);    // Solamente se permite la carga
    set_CHARGED_value(0);    // Batería sin cargar
    printf("La tension de la bateria es menor que la de 'bateria descargada'\n");
    printf("Se habilita el estado de emergencia\n");
    printf("DCHG_CHG = %i. Solamente permite la carga\n",
get_DCHG_CHG_value());
    printf("CHARGED = %i. Bateria descargada al maximo\n",
get_CHARGED_value());
    printf("Estado de emergencia = %i\n", emer);
}
else {
    emer = 0;

```

```

// printf("Sin errores en las lecturas de voltaje de descarga de la bateria, se sigue
verificando\n");
// }

printf("Se comprueba que la tension de la bateria esta dentro de los limites\n");
if (get_Total_V_value() >= get_Vcharged_value())
{

    set_DCHG_CHG_value(1);    // Solamente permite la descarga
    set_CHARGED_value(1);    // Bateria cargada incorrectamente
    printf("La tension de la bateria es mayor que la de 'bateria cargada'\n");
    printf("Se habilita el estado de emergencia\n");
    printf("DCHG_CHG = %i. Solamente permite la descarga\n",
get_DCHG_CHG_value());
    printf("CHARGED = %i. Bateria cargada al maximo\n", get_CHARGED_value());
// }

printf("Tension de bateria medida = %li < %li Tension maxima carga?\n",
get_Total_V_value(),battery_info.SOA.sec_tension );

if (emer == 1) {
    EMER_LED = 1; // Se enciende el LED de emergencia
    printf("Como ha habido algun error en la verificacion, se enciende el LED de
emergencia\n");
}
else {
    EMER_LED = 0;    // Se deshabilita el LED de emergencia
    printf("Verificacion completa y sin errores\n");
}
printf("Error final de verificacion = %i\n", error);
return error;
}

////////////////////////////////////
// Estados de las barras de LED's
// 0 = Deshabilitar la barra

```

```

// 1 = Ver estado de carga [SOC]
// 2 = Cargando, barra moviéndose
////////////////////////////////////

void LED_stripe (short int activate)
{
switch (activate) // Funciones de la barra LED
{
case 0:    // Barra apagada
    LATFbits.LATF4 = 0; // Parte alta de la barra de leds
    LATFbits.LATF5 = 0; // Parte baja de la barra de leds
    break;

case 1:    // Ver estado de carga
    if (get_Total_V_value()>get_charge_75_value()) LATFbits.LATF5 = 1;
    else LATFbits.LATF4 = 0;          // Parte baja de la barra de leds

    if (get_Total_V_value()>get_charge_100_value()) LATFbits.LATF5 = 1;
    else LATFbits.LATF4 = 0;          // Parte alta de la barra de leds

    set_counter_T_value(0);          // Se inicializa
    break;

case 2:    // Barra deslizándose
    if (get_counter_T_value()<10)
    {
        LATFbits.LATF5 = 0; // Parte baja de la barra de leds
        LATFbits.LATF4 = 0; // Parte alta de la barra de leds
    }

    if ((get_counter_T_value()<10) && (get_counter_T_value()<20))
    {
        LATFbits.LATF5 = 1; // Parte baja de la barra de leds
    }
}
}

```

```

    LATFbits.LATF4 = 0;    // Parte alta de la barra de leds
}

if ((get_counter_T_value()<20) && (get_counter_T_value()<30))
{
    LATFbits.LATF5 = 1;    // Parte baja de la barra de leds
    LATFbits.LATF4 = 1;    // Parte alta de la barra de leds
}
if (get_counter_T_value()==30) set_counter_T_value(0);

    increase_counter_T_value();
break;

default: set_counter_T_value(0); // Por defecto se resetea el contador
}
}

////////////////////////////////////
// Rutina de activación de los MOSFET's pertinentes
////////////////////////////////////

void MOSFET_enable (void)
{
    printf("Entramos a MOSFET_enable!\n");
    switch (get_DCHG_CHG_value()) {
        case 0:          // No se permite nada
            DCHG = 0;
            CHG = 0;
            printf("La variable DCHG es %i\n",DCHG);
            printf("La variable CHG es %i\n",CHG);
            printf("Por tanto, no se permite nada\n");
            break;

```

```

case 1:          // Solamente permite la descarga
    DCHG = 1;
    CHG = 0;
    printf("La variable DCHG es %i\n",DCHG);
    printf("La variable CHG es %i\n",CHG);
    printf("Por tanto, solo se permite la descarga\n");
break;

case 2:          //Solamente se permite la carga
    DCHG = 0;
    CHG = 1;
    printf("La variable DCHG es %i\n",DCHG);
    printf("La variable CHG es %i\n",CHG);
    printf("Por tanto, solo se permite la carga\n");
break;

case 3:          // Se permite todo
    DCHG = 1;
    CHG = 1;
    printf("La variable DCHG es %i\n",DCHG);
    printf("La variable CHG es %i\n",CHG);
    printf("Por tanto, se permite tanto la carga como la descarga\n");
break;

default:        // No se permite nada
    DCHG = 0;
    CHG = 0;
    printf("La variable DCHG es %i\n",DCHG);
    printf("La variable CHG es %i\n",CHG);
    printf("Por tanto, no se permite nada\n");
}
}

```

```

////////////////////////////////////
// Modo descarga
////////////////////////////////////

void DCHG_mode (void)
{
    set_DCHG_CHG_value(1);      // Solamente descarga
    set_CHARGED_value(0);      // No está al 100%
    calib_disable ();          // Se desactiva el balanceo de celdas
    LED_stripe(1);
}

////////////////////////////////////
// Modo carga
////////////////////////////////////

void CHG_mode (void)
{
    Tbattery battery_info;
    set_DCHG_CHG_value(2);      // Solamente carga
    printf("Se habilita la medicion para controlar el desequilibrio\n");
    cell_calib (battery_info.chg_balancing); //Habilitar equilibrado de celdas
    if (get_Total_V_value())>=get_Vcharged_value())
    {
        set_DCHG_CHG_value(1);    // Solamente descarga
        set_CHARGED_value(1);
        if (get_Current_value())>battery_info.min_chg_current)
        set_CHARGED_value(2); //Cargada correctamente
    }
    else set_CHARGED_value(0);    // Se elimina el fallo de descarga máxima de
    una celda
    LED_stripe (2);              // Barra de led moviéndose
    printf("Se enciende la barra de LED's con movimiento continuo indicando
    carga\n");
}

```

```

}

////////////////////////////////////
// Modo espera
////////////////////////////////////

void WAITING_mode (void)
{
    Tbattery battery_info;
    LED_stripe (1);          // Activar la barra con la indicación del SOC
    if (((get_Total_V_value())<get_discharged_battery_value()) ||
        (get_CHARGED_value()==4) && (get_CHARGED_value()!=3))
        // Batería descargada
    {
        set_DCHG_CHG_value(2);          // Solamente se permite la carga
        calib_disable ();              // Deshabilita el balanceo de las celdas
    }

    if (((get_CHARGED_value()==1) || (get_CHARGED_value()==2)) &&
        (get_Total_V_value())>get_Vrate_value()*1.04))
        // Está cargada, equilibrando y permitiendo la descarga
    {
        set_DCHG_CHG_value(1);          // Solamente se permite la descarga
        cell_calib (battery_info.stb_balancing);
    }

    if (((get_Total_V_value())>get_discharged_battery_value()) &&
        (get_Total_V_value())<get_Vrate_value()*1.04) && (get_CHARGED_value()!=4) &&
        (get_CHARGED_value()!=3))
        // Cargada, tensión de flotación y se permiten ambas cosas
    {
        set_DCHG_CHG_value(2);          // Se permite todo
        calib_disable ();              // Se deshabilita el balanceo
    }
}

```

```

/*
 * File: initialize.c
 * Author: Ahullonet
 *
 * Created on July 19, 2020, 5:08 PM
 */

#include <p24FJ256DA210.h>
#include "auxiliar.h"
#include <xc.h>
#include "I2C.h"
#include "initialize.h"
#include "structs.h"
#include "balancing.h"
#include <stdio.h>

////////////////////////////////////
// Inicialización del dispositivo de gestión (BMS)
////////////////////////////////////
unsigned long discharged_battery;
unsigned long Vrate;
int MA,zero_calib, DCHG_CHG, CHARGED;
unsigned long charge_100, charge_75;
long int Vcharged, counter_T;
double Temper1, Temper2, Temper3, Temper4;

Tfeedback Response(char RX_data);

void Initialize (void)
{
    Tbattery battery_info;

```

```

int n_used_total_cells;

config_oscil();      // Configuración del oscilador
printf("Se ha configurado el oscilador principal\n");
config_ports();     // Configuración de los puertos
printf("Se han configurado los puertos conectados al microcontrolador\n");
config_UART();      // Configuración de los UART a 19200 bps y 1bit stop
printf("Se han configurado los puertos y dispositivos serie\n");
I2C_init(F_100KHz); // Configuración de I2C a la velocidad estándar de 100
KHz
printf("Se ha configurado la comunicacion I2C\n");
LATFbits.LATF7 = 1; // Se deshabilita el reset de celdas
LATFbits.LATF3 = 0; // Se deshabilita el Clock del ADC
LATBbits.LATB10 = 0; // Se deshabilita el enable de celdas

LATDbits.LATD8 = 0; // Se deshabilitan los multiplexores
LATDbits.LATD11 = 1; // Se deshabilita la señal WR
LATDbits.LATD10 = 1; // CONVST a 1
LATDbits.LATD9 = 0; // CLK a 0
LATDbits.LATD0 = 1; // RD a 1

set_DCHG_CHG_value(2); // Para que calibre el sensor de corriente
// printf("El valor de la variable DCHG_CHG es %i\n", get_DCHG_CHG_value());
MOSFET_enable();

FAN = 0; // Deshabilita el ventilador
printf("Ventilador deshabilitado\n");
EMER_LED = 0; // Deshabilita el LED de emergencia
printf("LED de emergencia apagado\n");

LED_stripe(0); // Deshabilita la barra LED

```

```

set_MA_value(1);          // Si MA = 1, es MASTER Si MA = 0, es Slave

printf("La variable MA es %i\n",MA);

set_CHARGED_value(1);// Para indicar que no está cargada o que no ha habido
carga previa

printf("La variable CHARGED = %i indica que la batería no se ha descargado
después de cargar o que todavía carga\n", get_CHARGED_value());

printf("SE ESTABLECEN LOS PARAMETROS DE LA BATERIA A GESTIONAR\n");

//Inicialización de battery_info por defecto

battery_info.n_used_blockA_cells = 8;

printf("Se determina que el número de celdas en el bloque A es
%i\n",battery_info.n_used_blockA_cells);

battery_info.n_used_blockB_cells = 8;

printf("Se determina que el número de celdas en el bloque B es
%i\n",battery_info.n_used_blockB_cells);

battery_info.max_chg_current = -12500;

printf("Se determina la corriente máxima de carga en
%i\n",battery_info.max_chg_current);

battery_info.min_chg_current = -500;

printf("Se determina la corriente mínima de carga en
%i\n",battery_info.min_chg_current);

battery_info.peak_chg_current = -25000;

printf("Se determina la corriente de pico de carga en
%i\n",battery_info.peak_chg_current);

battery_info.max_dchg_current = 35000;

printf("Se determina la corriente máxima de descarga en
%i\n",battery_info.max_dchg_current);

battery_info.peak_dchg_current = 50000;

printf("Se determina el pico de corriente de carga en
%i\n",battery_info.peak_chg_current);

battery_info.chg_balancing = 1.007;

printf("Se determina el rango de desequilibrio en la carga en
%f\n",battery_info.chg_balancing);

battery_info.stb_balancing = 1.005;

printf("Se determina el rango de desequilibrio una vez cargado en
%f\n",battery_info.stb_balancing);

```

```

// Cálculos para el soc
n_used_total_cells = battery_info.n_used_blockA_cells +
battery_info.n_used_blockB_cells;

printf("Numero de celdas conectadas = %d\n", n_used_total_cells);

// Número de celdas conectadas
set_discharged_battery_value((long)n_used_total_cells *
(long)battery_info.SOA.max_dchg_tension);

// Tensión de descarga
set_Vrate_value((long)n_used_total_cells * (long)battery_info.SOA.rate_tension);

// Tensión nominal de batería
set_charge_100_value(get_Vrate_value() + (150*n_used_total_cells));

// Carga al 100% para los LED's
set_charge_75_value(get_Vrate_value() - (150*n_used_total_cells));

// Carga al 75% para los LED's
set_Vcharged_value ((long)n_used_total_cells *
(long)battery_info.SOA.max_chg_tension);

// Tensión mínima de la batería a plena carga

battery_info.SOA.t_max = 55.0;

printf("Se determina la temperatura maxima de funcionamiento en
%f\n",battery_info.SOA.t_max);

battery_info.SOA.sec_tension = 3650;

printf("Se determina la tension maxima de funcionamiento en
%i\n",battery_info.SOA.sec_tension);

battery_info.SOA.max_chg_tension = 3600;

printf("Se determina la tension maxima de carga en
%i\n",battery_info.SOA.max_chg_tension);

battery_info.SOA.rate_tension = 3200;

printf("Se determina la tension nominal de la celda en
%i\n",battery_info.SOA.rate_tension);

battery_info.SOA.max_dchg_tension = 2500;

printf("Se determina la tension maxima de descarga en
%f\n",battery_info.SOA.t_max);

set_counter_T_value(0); // Se reinicia el contador
set_Temper1_value((double) 25.0);

```

```

set_Temper2_value((double) 25.0);
set_Temper3_value((double) 25.0);
set_Temper4_value((double) 25.0);
// Se inicializa a la temperatura ambiente
printf("Se establece la temperatura ambiente en los sensores de temperatura, es
%2f\n",Temper1);
set_zero_calib_value(4090);           // Valor por defecto

if (get_MA_value() == 0)             // Si está como esclavo, RB0 y RB9 son
entradas
{
    ANSBbits.ANSB0 = DIGITAL;        // RB0 (DCHG para MA/SL)
    TRISBbits.TRISB0 = INPUT;

    ANSBbits.ANSB9 = DIGITAL;        // RB9 (CHG para MA/SL)
    TRISBbits.TRISB9 = INPUT;
}
calib_disable ();    // Se deshabilita el equilibrado de las celdas
ADC_ch_enable ();   // Habilitar los canales del ADC
printf("Se habilitan los canales 1, 2 y 3 del ADC interno\n");
read_data ();       // Se leen todos los datos para calibrar el sensor de corriente
set_DCHG_CHG_value(3);    // Se permite todo
}

////////////////////////////////////
// Configuración del oscilador
////////////////////////////////////

void config_oscil (void)
{
// Configuración de los LATEbits

OSCCONbits.COSC = 0b001;

```

```

// 001 = Fast RC Oscillator with Postscaler and PLL module (FRCPLL) 4MHz
OSCCONbits.SOSCEN = 0;
// 0 = Disable Secondary Oscillator
CLKDIVbits.RCDIV = 0;
// Así encendemos el oscilador primario (COSC), y apagamos el segundo (SOSC)
}

```

```

////////////////////////////////////

```

```

// Configuración de los puertos

```

```

////////////////////////////////////

```

```

void config_ports (void)

```

```

{

```

```

// TRIS -> 1 = INPUT 2 = OUTPUT;;; ANS -> 1 = ANALOG 0 = DIGITAL;;;

```

```

// ODC 1 = OPEN_DRAIN

```

```

// Se deshabilitan los comparadores

```

```

CM1CONbits.CEN = 0; // Comparador 1 deshabilitado

```

```

CM2CONbits.CEN = 0; // Comparador 2 deshabilitado

```

```

CM3CONbits.CEN = 0; // Comparador 3 deshabilitado

```

```

// Se deshabilita el USB para poder usar los pines G2 y G3

```

```

U1CON = 0x0000;

```

```

U1CNFG2bits.UTRDIS = 1;

```

```

U1CNFG2bits.UVCMPDIS = 1;

```

```

U1OTGCONbits.VBUSDIS = 0;

```

```

U1OTGCONbits.VBUSCHG = 1;

```

```

U1OTGCONbits.OTGEN = 0;

```

```

U1OTGCONbits.VBUSON = 0;

```

```

U1OTGCONbits.DMPULDWN = 0;

```

```

U1OTGCONbits.DPPULDWN = 0;

```

```

U1OTGCONbits.DMPULUP = 0;

```

```

U1OTGCONbits.DPPULUP = 0;

```

```

U1PWRCbits.USBPWR = 0;

TRISEbits.TRISE5 = OUTPUT; // RE5 (E2)
TRISEbits.TRISE6 = OUTPUT; // RE6 (E1)
TRISEbits.TRISE7 = OUTPUT; // RE7

ANSGbits.ANSG6 = DIGITAL;
TRISGbits.TRISG6 = OUTPUT; // RG6

TRISGbits.TRISG8 = INPUT; // RG8 RX
TRISGbits.TRISG9 = OUTPUT; // RG9 TX

ANSBbits.ANSB5 = DIGITAL;
TRISBbits.TRISB5 = OUTPUT; // RB5 (T1)
ODCBbits.ODB5 = OPEN_DRAIN;

ANSBbits.ANSB4 = DIGITAL;
TRISBbits.TRISB4 = OUTPUT; // RB4 (T2)
ODCBbits.ODB4 = OPEN_DRAIN;

ANSBbits.ANSB3 = DIGITAL;
TRISBbits.TRISB3 = OUTPUT; // RB3 (T3)
ODCBbits.ODB3 = OPEN_DRAIN;

ANSBbits.ANSB2 = DIGITAL;
TRISBbits.TRISB2 = OUTPUT; // RB2 (T4)
ODCBbits.ODB2 = OPEN_DRAIN;

ANSBbits.ANSB0 = DIGITAL;
TRISBbits.TRISB0 = OUTPUT; // RB0 (DCHG para MA/SL)

ANSBbits.ANSB8 = DIGITAL;
TRISBbits.TRISB8 = INPUT; // RB8 (Apagado)

```

```
ANSBbits.ANSB9 = DIGITAL;  
TRISBbits.TRISB9 = OUTPUT; // RB9 (CHG para MA/SL)
```

```
ANSBbits.ANSB10 = DIGITAL;  
TRISBbits.TRISB10 = OUTPUT; // RB10
```

```
ANSBbits.ANSB11 = DIGITAL;  
TRISBbits.TRISB11 = OUTPUT; // RB11
```

```
ANSBbits.ANSB12 = DIGITAL;  
TRISBbits.TRISB12 = OUTPUT; // RB12
```

```
ANSBbits.ANSB13 = DIGITAL;  
TRISBbits.TRISB13 = OUTPUT; // RB13
```

```
ANSBbits.ANSB14 = DIGITAL;  
TRISBbits.TRISB14 = OUTPUT; // RB14
```

```
ANSBbits.ANSB15 = DIGITAL;  
TRISBbits.TRISB15 = OUTPUT; // RB15
```

```
TRISFbits.TRISF4 = OUTPUT; // RF4 SDA  
TRISFbits.TRISF5 = OUTPUT; // RF5 SCL
```

```
TRISFbits.TRISF3 = OUTPUT; // RF3
```

```
TRISFbits.TRISF7 = OUTPUT; // RF7
```

```
TRISGbits.TRISG3 = OUTPUT; // RG3  
TRISGbits.TRISG2 = OUTPUT; // RG2
```

```
ANSGbits.ANSG8 = DIGITAL;
```

```

TRISGbits.TRISG8 = OUTPUT; // RG8

ANSGbits.ANSG9 = DIGITAL;
TRISGbits.TRISG9 = INPUT; // RG9

RPOR9bits.RP19R = 3; // UART1RX en pin RP27 (RG9)
RPINR18bits.U1RXR = 27; // UART1TX en pin RP19 (RG8)

TRISDbits.TRISD0 = OUTPUT; // RD0
TRISDbits.TRISD8 = OUTPUT; // RD8
TRISDbits.TRISD9 = OUTPUT; // RD9
TRISDbits.TRISD10 = OUTPUT; // RD10
TRISDbits.TRISD11 = OUTPUT; // RD11
TRISCBits.TRISC13 = INPUT; // RC13
TRISCBits.TRISC14 = INPUT; // RC14
TRISCBits.TRISC15 = OUTPUT; // RC15

TRISDbits.TRISD1 = INPUT; // RD1
TRISDbits.TRISD2 = INPUT; // RD2
TRISDbits.TRISD3 = INPUT; // RD3
TRISDbits.TRISD4 = INPUT; // RD4
TRISDbits.TRISD5 = INPUT; // RD5
ANSDbits.ANSD6 = DIGITAL;
TRISDbits.TRISD6 = INPUT; // RD6
ANSDbits.ANSD7 = DIGITAL;
TRISDbits.TRISD7 = INPUT; // RD7

ANSFbits.ANSF0 = DIGITAL;
TRISFbits.TRISF0 = INPUT; // RF0
TRISFbits.TRISF1 = INPUT; // RF1
TRISFbits.TRISF3 = OUTPUT; // RF3
TRISFbits.TRISF7 = OUTPUT; // RF7

```

```

TRISEbits.TRISE0 = INPUT; // RE0
TRISEbits.TRISE1 = INPUT; // RE1
TRISEbits.TRISE2 = INPUT; // RE2
TRISEbits.TRISE3 = INPUT; // RE3
TRISEbits.TRISE4 = INPUT; // RE4

// NOT USED

TRISGbits.TRISG7 = OUTPUT; // RG7
LATGbits.LATG7 = 0; // RG7 puesta a 0

ANSBbits.ANSB1 = DIGITAL;
TRISBbits.TRISB1 = OUTPUT; // RB1
LATBbits.LATB1 = 0; // RB1 puesta a 0
}

unsigned long get_discharged_battery_value () {
    return discharged_battery;
}

void set_discharged_battery_value (unsigned long valor) {
    discharged_battery = valor;
}

unsigned long get_Vrate_value () {
    return Vrate;
}

void set_Vrate_value (unsigned long valor) {
    Vrate = valor;
}

int get_MA_value () {

```

```

    return MA;
}

void set_MA_value (int valor) {
    MA = valor;
}

int get_zero_calib_value () {
    return zero_calib;
}

void set_zero_calib_value (int valor) {
    zero_calib = valor;
}

int get_DCHG_CHG_value () {
    return DCHG_CHG;
}

void set_DCHG_CHG_value (int valor) {
    DCHG_CHG = valor;
}

int get_CHARGED_value () {
    return CHARGED;
}

void set_CHARGED_value (int valor) {
    CHARGED = valor;
}

unsigned long get_charge_100_value () {
    return charge_100;
}

```

```
}
```

```
void set_charge_100_value (unsigned long valor) {  
    charge_100 = valor;  
}
```

```
unsigned long get_charge_75_value () {  
    return charge_75;  
}
```

```
void set_charge_75_value (unsigned long valor) {  
    charge_75 = valor;  
}
```

```
long int get_Vcharged_value () {  
    return Vcharged;  
}
```

```
void set_Vcharged_value (long int valor) {  
    Vcharged = valor;  
}
```

```
long int get_counter_T_value () {  
    return counter_T;  
}
```

```
void set_counter_T_value (long int valor) {  
    counter_T = valor;  
}
```

```
void increase_counter_T_value () {  
    counter_T++;  
}
```

```
double get_Temper1_value () {
    return Temper1;
}

void set_Temper1_value (double valor) {
    Temper1 = valor;
}

double get_Temper2_value () {
    return Temper2;
}

void set_Temper2_value (double valor) {
    Temper2 = valor;
}

double get_Temper3_value () {
    return Temper3;
}

void set_Temper3_value (double valor) {
    Temper3 = valor;
}

double get_Temper4_value () {
    return Temper4;
}

void set_Temper4_value (double valor) {
    Temper4 = valor;
}
```

```

/*
 * File: initialize.h
 * Author: Ahullonet
 *
 * Created on September 9, 2020, 7:01 PM
 */

#ifndef INITIALIZE_H
#define INITIALIZE_H

////////////////////////////////////
// Declaración de tipos
////////////////////////////////////

#define N_TOTAL_CELLS 16 // Máximo de 16 celdas
#define N_BLOCK_CELLS 8 // Máximo de 8 celdas por bloque
#define T_MAX_PEAK 100 // Tiempo máx de pico
#define T_FAN 50.0 // Por encima de 50° se enciende
#define POSITIVE_SENS_A 500 // Por encima empieza la descarga
#define POSITIVE_SENS_B 350 // Por debajo termina la descarga
#define NEGATIVE_SENS_A -500 // Por debajo empieza la carga
#define NEGATIVE_SENS_B -350 // Por arriba termina la descarga

// TRIS
#define OUTPUT 0
#define INPUT 1

// ANS
#define ANALOG 1
#define DIGITAL 0

// ODC

```

```

#define OPEN_DRAIN 1
#define AVERAGE 50          // Lecturas necesarias para hacer la media

// #define SW_OFF  PORTBbits.RB8 // Switch OFF = 1 y ON = 0
#define CHG      LATEbits.LATE6 // Carga
#define DCHG     LATEbits.LATE5 // Descarga
#define FAN      LATEbits.LATE7 // Ventilador
#define EMER_LED LATGbits.LATG6 // LED de emergencia

////////////////////////////////////
// Declaración de tipos
////////////////////////////////////

unsigned long get_discharged_battery_value();
void set_discharged_battery_value(unsigned long valor);
unsigned long get_Vrate_value();
void set_Vrate_value(unsigned long valor);
int get_MA_value();
void set_MA_value(int valor);
int get_zero_calib_value();
void set_zero_calib_value(int valor);
int get_DCHG_CHG_value();
void set_DCHG_CHG_value(int valor);
int get_CHARGED_value();
void set_CHARGED_value(int valor);
unsigned long get_charge_100_value();
void set_charge_100_value(unsigned long valor);
unsigned long get_charge_75_value();
void set_charge_75_value(unsigned long valor);
long int get_Vcharged_value();
void set_Vcharged_value(long int valor);
long int get_counter_T_value();
void set_counter_T_value(long int valor);

```

```
void increase_counter_T_value();
double get_Temper1_value();
void set_Temper1_value(double valor);
double get_Temper2_value();
void set_Temper2_value(double valor);
double get_Temper3_value();
void set_Temper3_value(double valor);
double get_Temper4_value();
void set_Temper4_value(double valor);

#endif      /* INITIALIZE_H */
```

```

/*
 * File: readsymux.c
 * Author: Ahullonet
 *
 * Created on August, 2020, 3:12 PM
 */

#define FOSC (32000000)
#define FCY (FOSC/2)

#include <xc.h>
#include <libpic30.h>
#include "auxiliar.h"
#include "readsymux.h"
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "structs.h"
#include "initialize.h"
#include "balancing.h"
#include "ADC.h"

////////////////////////////////////
// Criterio de selección de canal del multiplexor
////////////////////////////////////
unsigned long long int Va, Vb;
long int Total_V, Current;

int channel_sel_MUX (unsigned int channel)
{
    int error = 0;

    LATDbits.LATD8 = 1;    // Se habilitan los multiplexores

```

```
__delay_us(1);      // Retraso
```

```
switch (channel)
```

```
{
```

```
  case 0:
```

```
  {
```

```
    LATCbits.LATC15 = 0;
```

```
    LATGbits.LATG2 = 0;
```

```
    LATGbits.LATG3 = 0;
```

```
  }
```

```
  break;
```

```
  case 1:
```

```
  {
```

```
    LATCbits.LATC15 = 1;
```

```
    LATGbits.LATG2 = 0;
```

```
    LATGbits.LATG3 = 0;
```

```
  }
```

```
  break;
```

```
  case 2:
```

```
  {
```

```
    LATCbits.LATC15 = 0;
```

```
    LATGbits.LATG2 = 1;
```

```
    LATGbits.LATG3 = 0;
```

```
  }
```

```
  break;
```

```
  case 3:
```

```
  {
```

```
    LATCbits.LATC15 = 1;
```

```
    LATGbits.LATG2 = 1;
```

```
    LATGbits.LATG3 = 0;
```

```
}  
break;  
  
case 4:  
{  
    LATCbits.LATC15 = 0;  
    LATGbits.LATG2 = 0;  
    LATGbits.LATG3 = 1;  
}  
break;  
  
case 5:  
{  
    LATCbits.LATC15 = 1;  
    LATGbits.LATG2 = 0;  
    LATGbits.LATG3 = 1;  
}  
break;  
  
case 6:  
{  
    LATCbits.LATC15 = 0;  
    LATGbits.LATG2 = 1;  
    LATGbits.LATG3 = 1;  
}  
break;  
  
case 7:  
{  
    LATCbits.LATC15 = 1;  
    LATGbits.LATG2 = 1;  
    LATGbits.LATG3 = 1;  
}
```

```

break;

default: error = 1;
// En caso de que la variable no sea ninguna de las anteriores
} return error;
}

////////////////////////////////////
// Lectura de los convertidores
////////////////////////////////////

void read_data (void)
{
int i, j;
unsigned long long int aver_Va[AVERAGE];
unsigned long long int aver_Vb[AVERAGE];
unsigned long long int aver_I[AVERAGE];
Tbattery battery_info;
Tfeedback Response(char RX_data);

i = 0;
for (i=0;i<N_BLOCK_CELLS;i++)
{
// printf("Se procede a medir la celda %i\n", i);
channel_sel_MUX (i);
// printf("Se habilita el canal %i del multiplexor para poder medir\n", i);
USB_comm (); // Se revisa si hay petición de datos
// printf("Mediante el ADC, se realizan 50 medidas del voltaje total del bloque A,
B y la I instantanea\n");
// printf("Se calcula la media para mejorar la calidad de las lecturas\n");

for (j=0;j<(AVERAGE);j++)
{

```

```

    read_ADC();
//   printf("Hemos salido del read ADC\n");
    aver_Va[j] = get_Va_value();
//   printf("Va en posicion %i = %llu\n", j, aver_Va[j]);
    aver_Vb[j] = get_Vb_value();
//   printf("Vb en posicion %i = %i\n", j, get_Vb_value());
    aver_I[j] = get_I_value();
}
set_Va_value(0);
set_Vb_value(0);
set_I_value(0);

for (j=0;j<(AVERAGE);j++) // Cálculo de la media de las lecturas
{
    set_Va_value(get_Va_value() + aver_Va[j]);
    set_Vb_value(get_Vb_value() + aver_Vb[j]);
    set_I_value(get_I_value() + aver_I[j]);
}
//   printf("Valor total de las celdas del bloque A es %llu\n", get_Va_value());
//   printf("Valor total de las celdas del bloque B es %llu\n", get_Vb_value());
//   printf("Valor total de la corriente es %llu\n", get_I_value());
set_Va_value(get_Va_value()/AVERAGE);
set_Vb_value(get_Vb_value()/AVERAGE);
set_I_value(get_I_value()/AVERAGE);
set_Current_value(get_I_value());
//   printf("Vmedia Celda %i del bloque A es %llu\n", i, get_Va_value());
//   printf("Vmedia Celda %i del bloque B es %llu\n", i, get_Vb_value());
//   printf("La media de la corriente es %llu\n", get_I_value());

if ((get_DCHG_CHG_value() == 0)) set_zero_calib_value(get_I_value());
//   set_Current_value((get_I_value() - get_zero_calib_value()));
//   printf("Corriente = %li\n", get_Current_value());
//   OA = 4090 y m = (15000+15000)/(5052-3105) = 30000/1946 = 15.41mA

```

```

    set_voltage_cells_pose(((get_Va_value()*1.21)-700),i);
//   printf("Voltaje de la celda %i = %li\n", i, get_voltage_cells_pose(i));
    // m = (4200-1800)/(3974-2064) = 2400/1909 = 1.25mV
    set_voltage_cells_pose(((get_Vb_value()*1.21)-693),N_BLOCK_CELLS +i);
//   printf("Voltaje de la celda %i = %li\n", (N_BLOCK_CELLS +i),
get_voltage_cells_pose(N_BLOCK_CELLS +i));
    // m = (4200-1800)/(3974-2064) = 2400/1909 = 1.25mV
}
// Cálculo de la tensión total de batería

for (i=0; i<battery_info.n_used_blockA_cells;i++)
{
//   printf("Entramos en el for A para medir el voltaje total\n");
    set_Total_V_value(get_voltage_cells_pose(i) + get_Total_V_value());
}
for (i=0; i<battery_info.n_used_blockB_cells;i++)
{
//   printf("Entramos en el for B para medir el voltaje total\n");
    set_Total_V_value(get_voltage_cells_pose(N_BLOCK_CELLS+i) +
get_Total_V_value());
}
printf("Corriente = %li\n", get_Current_value());
printf("Tension total de la bateria = %li\n", get_Current_value());
}

long int get_Current_value () {
    return Current;
}

void set_Current_value (long int valor) {
    Current = valor;
}

long int get_Total_V_value () {

```

```
    return Total_V;
}

void set_Total_V_value (long int valor) {
    Total_V = valor;
}

unsigned long long int get_Va_value () {
    return Va;
}

void set_Va_value (unsigned long long int valor) {
    Va = valor;
}

unsigned long long int get_Vb_value () {
    return Vb;
}

void set_Vb_value (unsigned long long int valor) {
    Vb = valor;
}
```

```

/*
 * File: readsymux.h
 * Author: Ahullonet
 *
 * Created on August, 2020, 8:32 PM
 */

#ifndef READSYMUX_H
#define READSYMUX_H

#define N_TOTAL_CELLS 16 // Máximo de 16 celdas
#define N_BLOCK_CELLS 8 // Máximo de 8 celdas por bloque
#define T_MAX_PEAK 100 // Tiempo máx de pico
#define T_FAN 50.0 // Por encima de 50° se enciende
#define POSITIVE_SENS_A 500 // Por encima empieza la descarga
#define POSITIVE_SENS_B 350 // Por debajo termina la descarga
#define NEGATIVE_SENS_A -500 // Por debajo empieza la carga
#define NEGATIVE_SENS_B -350 // Por arriba termina la descarga

// TRIS
#define OUTPUT 0
#define INPUT 1

// ANS
#define ANALOG 1
#define DIGITAL 0

// ODC
#define OPEN_DRAIN 1
#define AVERAGE 50 // Lecturas necesarias para hacer la media

// #define SW_OFF PORTBbits.RB8 // Switch OFF = 1 y ON = 0
#define CHG LATEbits.LATE6 // Carga

```

```
#define DCHG    LATEbits.LATE5 // Descarga
#define FAN     LATEbits.LATE7 // Ventilador
#define EMER_LED LATGbits.LATG6 // LED de emergencia
```

```
long int get_Current_value();
void set_Current_value(long int valor);
long int get_Total_V_value();
void set_Total_V_value(long int valor);
unsigned long long int get_Va_value();
void set_Va_value(unsigned long long int valor);
unsigned long long int get_Vb_value();
void set_Vb_value(unsigned long long int valor);
```

```
#endif      /* READSYMUX_H */
```

```
/*
 * File: ADC.c
 * Author: Ahullonet
 *
 * Created on July 22, 2020, 4:23 PM
 */
```

```
#define FOSC (32000000)
#define FCY (FOSC/2)
//#include <p24FJ256DA210.h>
#include <xc.h>
#include <stdlib.h>
#include <libpic30.h>
#include "ADC.h"
#include "readsymux.h"
#include "structs.h"
#include "stdio.h"
//#include "auxiliar.h"
```

```
////////////////////////////////////
// Lectura del convertidor
////////////////////////////////////
long long int l;
Tbattery battery_info;
Tfeedback comm_info;
Tfeedback Response(char RX_data);

void read_ADC (void)
{
    int i = 0;
```

```

set_Va_value(0);
set_Vb_value(0);
set_I_value(0);

LATDbits.LATD0 = 1;    // Lectura deshabilitada
LATDbits.LATD10 = 0;   // CONVST a cero
__delay_us(8);        // Retraso para dar tiempo a conversión CONVST
LATDbits.LATD10 = 1;   // CONVST a nivel alto
__delay_us(1);        // Retraso para dar tiempo a conversión CONVST

LATDbits.LATD9 = 1;    // CLK a nivel alto

for (i=1; i<30; i++)   // Generación de 30 pulsos de reloj
{
    __delay_us(1);     // Espera
    LATDbits.LATD9 = 0; // CLK a nivel bajo
    __delay_us(1);     // Tiempo de espera
    LATDbits.LATD9 = 1; // CLK a nivel alto

    if (i == 16)
    {
        LATDbits.LATD0 = 0; // Se habilita la primera lectura
        __delay_us(1);
        set_Va_value((PORTDbits.RD1*8192) + (PORTDbits.RD2*4096) +
(PORTDbits.RD3*2048) +
(PORTDbits.RD4*1024) + (PORTDbits.RD5*512) + (PORTDbits.RD6*256) +
(PORTDbits.RD7*128) + (PORTFbits.RF0*64) + (PORTFbits.RF1*32) +
(PORTEbits.RE0*16) + (PORTEbits.RE1*8) + (PORTEbits.RE2*4) +
(PORTEbits.RE3*2) + (PORTEbits.RE4));
        LATDbits.LATD0 = 1; // Se deshabilita la lectura
//    printf("Se mide el voltaje total del bloque A. Va = %i\n", get_Va_value());
    }
}

```

```

if (i==19)
{
    LATDbits.LATD0 = 0;    // Se deshabilita la segunda lectura
    __delay_us(1);
    set_Vb_value((PORTDbits.RD1*8192) + (PORTDbits.RD2*4096) +
(PORTDbits.RD3*2048) +
    (PORTDbits.RD4*1024) + (PORTDbits.RD5*512) + (PORTDbits.RD6*256) +
    (PORTDbits.RD7*128) + (PORTFbits.RF0*64) + (PORTFbits.RF1*32) +
    (PORTEbits.RE0*16) + (PORTEbits.RE1*8) + (PORTEbits.RE2*4) +
    (PORTEbits.RE3*2) + (PORTEbits.RE4));
    LATDbits.LATD0 = 1;    // Se deshabilita a lectura
//    printf("Se mide el voltaje total del bloque B. Vb = %i\n", get_Vb_value());
}

if (i==22)
{
    set_I_value((PORTDbits.RD1*8192) + (PORTDbits.RD2*4096) +
(PORTDbits.RD3*2048) +
    (PORTDbits.RD4*1024) + (PORTDbits.RD5*512) + (PORTDbits.RD6*256) +
    (PORTDbits.RD7*128) + (PORTFbits.RF0*64) + (PORTFbits.RF1*32) +
    (PORTEbits.RE0*16) + (PORTEbits.RE1*8) + (PORTEbits.RE2*4) +
    (PORTEbits.RE3*2) + (PORTEbits.RE4));
    printf("Se mide el voltaje total del bloque A. Va = %f\n", get_Va_value());
    if (get_I_value()>8191) set_I_value(get_I_value()-16384);

    LATDbits.LATD0 = 1;    // Se deshabilita a lectura
}

}

__delay_us(1);
LATDbits.LATD9 = 0;      // CLK a nivel bajo
}

```

```

////////////////////////////////////
// Habilitación canales ADC
////////////////////////////////////

void ADC_ch_enable (void)
{
    TRISEbits.TRISE4 = OUTPUT;    // RE4
    TRISEbits.TRISE3 = OUTPUT;    // RE3
    TRISEbits.TRISE2 = OUTPUT;    // RE2
    TRISEbits.TRISE1 = OUTPUT;    // RE1
    TRISEbits.TRISE0 = OUTPUT;    // RE0
    TRISFbits.TRISF1 = OUTPUT;    // RF1
    TRISFbits.TRISF0 = OUTPUT;    // RF0
    TRISDbits.TRISD7 = OUTPUT;    // RD7

    // Se habilitan los siguientes canales

    LATEbits.LATE4 = 1;
    LATEbits.LATE3 = 1;
    LATEbits.LATE2 = 1;
    LATEbits.LATE1 = 0;
    LATEbits.LATE0 = 0;
    LATFbits.LATF1 = 0;
    LATFbits.LATF0 = 0;
    LATDbits.LATD7 = 0;
    __delay_us(10);

    //Se activa la señal de escritura
    LATDbits.LATD11 = 0;        // Señal WR habilitada
    __delay_us(5);
    LATDbits.LATD11 = 1;        // Señal WR deshabilitada

    TRISEbits.TRISE4 = INPUT;    // RE4

```

```
TRISEbits.TRISE3 = INPUT; // RE3
TRISEbits.TRISE2 = INPUT; // RE2
TRISEbits.TRISE1 = INPUT; // RE1
TRISEbits.TRISE0 = INPUT; // RE0
TRISFbits.TRISF1 = INPUT; // RF1
TRISFbits.TRISF0 = INPUT; // RF0
TRISDbits.TRISD7 = INPUT; // RD7
__delay_us(10);
}
```

```
long long int get_I_value () {
    return I;
}
```

```
void set_I_value (long long int valor) {
    I = valor;
}
```

```

/*
 * File:  ADC.h
 * Author: Ahullonet
 *
 * Created on September 10, 2020, 8:47 PM
 */

#ifndef ADC_H
#define    ADC_H

#define N_TOTAL_CELLS 16    // Máximo de 16 celdas
#define N_BLOCK_CELLS 8    // Máximo de 8 celdas por bloque
#define T_MAX_PEAK 100    // Tiempo máx de pico
#define T_FAN 50.0        // Por encima de 50° se enciende
#define POSITIVE_SENS_A 500    // Por encima empieza la descarga
#define POSITIVE_SENS_B 350    // Por debajo termina la descarga
#define NEGATIVE_SENS_A -500    // Por debajo empieza la carga
#define NEGATIVE_SENS_B -350    // Por arriba termina la descarga

// TRIS
#define OUTPUT 0
#define INPUT 1

// ANS
#define ANALOG 1
#define DIGITAL 0

// ODC
#define OPEN_DRAIN 1
#define AVERAGE 50    // Lecturas necesarias para hacer la media

// #define SW_OFF  PORTBbits.RB8 // Switch OFF = 1 y ON = 0
#define CHG  LATEbits.LATE6 // Carga

```

```
#define DCHG    LATEbits.LATE5 // Descarga
#define FAN     LATEbits.LATE7 // Ventilador
#define EMER_LED LATGbits.LATG6 // LED de emergencia

////////////////////////////////////
// Declaración de tipos
////////////////////////////////////

long long int get_I_value();
void set_I_value(long long int valor);

#endif      /* ADC_H */
```

```

/*
 * File:  balancing.c
 * Author: Ahullonet
 *
 * Created on July 22, 2020, 3:58 PM
 */

#define FOSC (32000000)
#define FCY (FOSC/2)

#include <xc.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <libpic30.h>
#include "structs.h"
#include "balancing.h"
#include "auxiliar.h"
#include "ADC.h"

////////////////////////////////////
// Balanceo de las celdas
////////////////////////////////////

////////////////////////////////////
// Se busca la celda con menos tensión y las celdas que
// están por encima del porcentaje del voltaje de carga
// si hay alguna en estos casos, se activa el equilibrado
////////////////////////////////////

char calib_state[2];
long int voltage_cells [N_TOTAL_CELLS];

```

```

void cell_calib (double percent) // Se busca la celda de menor tensión
{
    int i;
    long int stop;
    double lower;
    Tbattery battery_info;
    Tfeedback Response(char RX_data);

    set_calib_state_pose(0,0); // Se resetean los indicadores del calibrado del bloque
A
    set_calib_state_pose(0,1); // Se resetean los indicadores del calibrado del bloque
B

    lower = get_voltage_cells_pose(0);
    for (i=0;i<battery_info.n_used_blockA_cells;i++)
    {
        if (get_voltage_cells_pose(i)<lower) lower = get_voltage_cells_pose(i);
// printf("La celda medida con menos tension es la %i\n", i);
// printf("Y su voltaje es %i\n",get_voltage_cells_pose(i));

    }
    for (i=0;i<(battery_info.n_used_blockB_cells-2);i++)
    {
        if (get_voltage_cells_pose(N_BLOCK_CELLS)<lower) lower =
get_voltage_cells_pose(N_BLOCK_CELLS+i);
        stop = lower*percent; // Con esto se sabe cuál es la celda con menos tensión
// printf("La celda medida con menos tension es la %i\n", N_BLOCK_CELLS+i);
// printf("Y su voltaje es %i\n",get_voltage_cells_pose(N_BLOCK_CELLS+i));
        printf("La celda medida con menos tension es la %i\n", N_BLOCK_CELLS+i);
    }
}

```

```

for (i=0;i<battery_info.n_used_blockA_cells;i++)
{
    if ((get_voltage_cells_pose(i)>stop) ||
(get_voltage_cells_pose(i)>battery_info.SOA.max_chg_tension)){
        printf("Se habilita el calibrado del bloque A\n");
        calibrate (i,1);
    }
    else calibrate (i,0);
}
printf("Se habilita el calibrado del bloque B\n");
for (i=0;i<battery_info.n_used_blockB_cells-1;i++)
{
    if ((get_voltage_cells_pose(N_BLOCK_CELLS+i)>stop) ||
(get_voltage_cells_pose(N_BLOCK_CELLS+i)>battery_info.SOA.max_chg_tension)
) {

        calibrate(N_BLOCK_CELLS+(i-1),1);}
    else calibrate(N_BLOCK_CELLS+(i-1),0);
}
}

////////////////////////////////////
// Calibrado a la celda correspondiente
////////////////////////////////////

int calibrate (unsigned int cell_to_calib, unsigned int activate)
{
    int error = 0;
    if (activate == 1)
        LATBbits.LATB15 = 1;    // Dato "Activar" a escribir
    else LATBbits.LATB15 = 0; // Dato "Desactivar" a escribir

    switch (cell_to_calib) // Según la celda que haya que calibrar, se escoge el CASE

```

```
{
case 0:
    LATBbits.LATB11 = 0;
    LATBbits.LATB12 = 0;
    LATBbits.LATB13 = 0;
    LATBbits.LATB14 = 0;
    if (activate == 1) increase_calib_state_pose(1,0);
break;

case 1:
    LATBbits.LATB11 = 1;
    LATBbits.LATB12 = 0;
    LATBbits.LATB13 = 0;
    LATBbits.LATB14 = 0;
    if (activate == 1) increase_calib_state_pose(2,0);
break;

case 2:
    LATBbits.LATB11 = 0;
    LATBbits.LATB12 = 1;
    LATBbits.LATB13 = 0;
    LATBbits.LATB14 = 0;
    if (activate == 1) increase_calib_state_pose(4,0);
break;

case 3:
    LATBbits.LATB11 = 1;
    LATBbits.LATB12 = 1;
    LATBbits.LATB13 = 0;
    LATBbits.LATB14 = 0;
    if (activate == 1) increase_calib_state_pose(8,0);
break;
```

case 4:

```
LATBbits.LATB11 = 0;  
LATBbits.LATB12 = 0;  
LATBbits.LATB13 = 1;  
LATBbits.LATB14 = 0;  
if (activate == 1) increase_calib_state_pose(16,0);
```

break;

case 5:

```
LATBbits.LATB11 = 1;  
LATBbits.LATB12 = 0;  
LATBbits.LATB13 = 1;  
LATBbits.LATB14 = 0;  
if (activate == 1) increase_calib_state_pose(32,0);
```

break;

case 6:

```
LATBbits.LATB11 = 0;  
LATBbits.LATB12 = 1;  
LATBbits.LATB13 = 1;  
LATBbits.LATB14 = 0;  
if (activate == 1) increase_calib_state_pose(64,0);
```

break;

case 7:

```
LATBbits.LATB11 = 1;  
LATBbits.LATB12 = 1;  
LATBbits.LATB13 = 1;  
LATBbits.LATB14 = 0;  
if (activate == 1) increase_calib_state_pose(128,0);
```

break;

case 8:

```
LATBbits.LATB11 = 0;
LATBbits.LATB12 = 0;
LATBbits.LATB13 = 0;
LATBbits.LATB14 = 1;
if (activate == 1) increase_calib_state_pose(1,1);
printf("Se habilita el MOSFET de la celda 8\n");
break;
```

case 9:

```
LATBbits.LATB11 = 1;
LATBbits.LATB12 = 0;
LATBbits.LATB13 = 0;
LATBbits.LATB14 = 1;
if (activate == 1) increase_calib_state_pose(2,1);
printf("Se habilita el MOSFET de la celda 9\n");
break;
```

case 10:

```
LATBbits.LATB11 = 0;
LATBbits.LATB12 = 1;
LATBbits.LATB13 = 0;
LATBbits.LATB14 = 1;
if (activate == 1) increase_calib_state_pose(4,1);
printf("Se habilita el MOSFET de la celda 10\n");
break;
```

case 11:

```
LATBbits.LATB11 = 1;
LATBbits.LATB12 = 1;
LATBbits.LATB13 = 0;
LATBbits.LATB14 = 1;
if (activate == 1) increase_calib_state_pose(8,1);
printf("Se habilita el MOSFET de la celda 11\n");
```

```
break;
```

```
case 12:
```

```
    LATBbits.LATB11 = 0;
```

```
    LATBbits.LATB12 = 0;
```

```
    LATBbits.LATB13 = 1;
```

```
    LATBbits.LATB14 = 1;
```

```
    if (activate == 1) increase_calib_state_pose(16,1);
```

```
    printf("Se habilita el MOSFET de la celda 12\n");
```

```
break;
```

```
case 13:
```

```
    LATBbits.LATB11 = 1;
```

```
    LATBbits.LATB12 = 0;
```

```
    LATBbits.LATB13 = 1;
```

```
    LATBbits.LATB14 = 1;
```

```
    if (activate == 1) increase_calib_state_pose(32,1);
```

```
    printf("Se habilita el MOSFET de la celda 13\n");
```

```
break;
```

```
case 14:
```

```
    LATBbits.LATB11 = 0;
```

```
    LATBbits.LATB12 = 1;
```

```
    LATBbits.LATB13 = 1;
```

```
    LATBbits.LATB14 = 1;
```

```
    if (activate == 1) increase_calib_state_pose(64,1);
```

```
    printf("Se habilita el MOSFET de la celda 14\n");
```

```
break;
```

```
case 15:
```

```
    LATBbits.LATB11 = 1;
```

```
    LATBbits.LATB12 = 1;
```

```
    LATBbits.LATB13 = 1;
```

```

LATBbits.LATB14 = 1;
if (activate == 1) increase_calib_state_pose(128,1);
printf("Se habilita el MOSFET de la celda 15\n");
break;

default: error = 1; // En caso de que no sea ninguna variable, devuelve error
}

if (error == 0)
{
LATFbits.LATF3 = 1; // Se habilita la escritura
__delay_us (10); // Retraso para dejar tiempo a la escritura
LATFbits.LATF3 = 0; // Se deshabilita la escritura
}

LATBbits.LATB15 = 0; // Se desactiva la escritura

return error;
}
////////////////////////////////////
// Reset del calibrado o dejar las celdas sin calibrar
////////////////////////////////////

void calib_disable (void)
{
LATFbits.LATF7 = 0; // Se habilita el reset de celdas
__delay_us (10); // Duración del reset
LATFbits.LATF7 = 1; // Se deshabilita el reset de Celdas
set_calib_state_pose(0,0);

printf("La calibracion del bloque A se deshabilita, igual a
%i\n",get_calib_state_pose(0));

set_calib_state_pose(0,1);

printf("La calibracion del bloque B se deshabilita, igual a
%i\n",get_calib_state_pose(1));

```

```
}
```

```
void set_calib_state_pose(long int valor, int position) {  
    calib_state[position] = valor;  
}
```

```
char get_calib_state_pose(int position) {  
    return calib_state[position];  
}
```

```
void increase_calib_state_pose(long int valor, int position) {  
    calib_state[position] = calib_state[position] + valor;  
}
```

```
void set_voltage_cells_pose(long int valor, int position) {  
    voltage_cells[position] = valor;  
}
```

```
long int get_voltage_cells_pose(int position) {  
    return voltage_cells[position];  
}
```

```
long int divide_get_voltage_cells_pose(long int valor, int position) {  
    voltage_cells[position] = (voltage_cells[position] / valor);  
    return voltage_cells[position];  
}
```

```
long int subtraction_get_voltage_cells_pose(long int valor, int position) {  
    voltage_cells[position] = (voltage_cells[position] - valor);  
    return voltage_cells[position];  
}
```

```

/*
 * File:  balancing.h
 * Author: Ahullonet
 *
 * Created on September 11, 2020, 6:52 PM
 */

#ifndef BALANCING_H
#define    BALANCING_H

#define N_TOTAL_CELLS 16    // Máximo de 16 celdas
#define N_BLOCK_CELLS 8    // Máximo de 8 celdas por bloque
#define T_MAX_PEAK 100    // Tiempo máx de pico
#define T_FAN 50.0        // Por encima de 50° se enciende
#define POSITIVE_SENS_A 500 // Por encima empieza la descarga
#define POSITIVE_SENS_B 350 // Por debajo termina la descarga
#define NEGATIVE_SENS_A -500 // Por debajo empieza la carga
#define NEGATIVE_SENS_B -350 // Por arriba termina la descarga

// TRIS
#define OUTPUT 0
#define INPUT 1

// ANS
#define ANALOG 1
#define DIGITAL 0

// ODC
#define OPEN_DRAIN 1
#define AVERAGE 50    // Lecturas necesarias para hacer la media

// #define SW_OFF PORTBbits.RB8 // Switch OFF = 1 y ON = 0
#define CHG    LATEbits.LATE6 // Carga

```

```
#define DCHG    LATEbits.LATE5 // Descarga
#define FAN     LATEbits.LATE7 // Ventilador
#define EMER_LED LATGbits.LATG6 // LED de emergencia

void set_calib_state_pose(long int valor, int position);
char get_calib_state_pose(int position);
void increase_calib_state_pose(long int valor, int position);
void set_voltage_cells_pose(long int valor, int position);
long int get_voltage_cells_pose(int position);
long int divide_get_voltage_cells_pose(long int valor, int position);
int subtraction_voltage_cells_pose(long int valor, int position);

#endif      /* BALANCING_H */
```

```

/*
 * File: auxiliar.h
 * Author: Ahullonet
 *
 * Created on July 20, 2020, 5:45 PM
 */

////////////////////////////////////
// Declaración de constantes
////////////////////////////////////

#ifndef AUXILIAR_H
#define AUXILIAR_H

#define N_TOTAL_CELLS 16 // Máximo de 16 celdas
#define N_BLOCK_CELLS 8 // Máximo de 8 celdas por bloque
#define T_MAX_PEAK 100 // Tiempo máx de pico
#define T_FAN 50.0 // Por encima de 50° se enciende
#define POSITIVE_SENS_A 500 // Por encima empieza la descarga
#define POSITIVE_SENS_B 350 // Por debajo termina la descarga
#define NEGATIVE_SENS_A -500 // Por debajo empieza la carga
#define NEGATIVE_SENS_B -350 // Por arriba termina la descarga

// TRIS
#define OUTPUT 0
#define INPUT 1

// ANS
#define ANALOG 1
#define DIGITAL 0

// ODC
#define OPEN_DRAIN 1

```

```

#define AVERAGE 50          // Lecturas necesarias para hacer la media

// #define SW_OFF  PORTBbits.RB8 // Switch OFF = 1 y ON = 0
#define CHG      LATEbits.LATE6 // Carga
#define DCHG     LATEbits.LATE5 // Descarga
#define FAN      LATEbits.LATE7 // Ventilador
#define EMER_LED LATGbits.LATG6 // LED de emergencia

////////////////////////////////////
// DCHG_CHG = 0 No permite nada
// DCHG_CHG = 1 Solamente descarga
// DCHG_CHG = 2 Solamente carga
// DCHG_CHG = 3 Permite tanto la carga como la descarga
////////////////////////////////////
// CHARGED = 0 Descargada
// CHARGED = 1 Mal cargada
// CHARGED = 2 Bien cargada
// CHARGED = 3 Celda sobrecargada
// CHARGED = 4 Celda sobredescargada
////////////////////////////////////

////////////////////////////////////
// Declaración de funciones
////////////////////////////////////

void config_oscil(void);
void config_ports(void);
void read_ADC(void);
void read_data(void);
void ADC_ch_enable(void);
int channel_sel_MUX(unsigned int channel);
void cell_calib(double percent);
int calibrate(unsigned int cell_to_calib, unsigned int activate);

```

```
void calib_disable(void);
void all_disabled(void);
void init_Thermos(void);
void read_temp_Thermos(void);
void config_UART(void);
char RX(void);
int TX(char data);
int limit_verify(void);
int USB_comm(void);
void LED_stripe(short int activate);
void Initialize(void);
void state_verify(void);
void DCHG_mode(void);
void CHG_mode(void);
void WAITING_mode(void);
void MOSFET_enable(void);
void ADC_ch_enable (void);

#endif
```

```

/*
 * File: communications.c
 * Author: Ahullonet
 *
 * Created on July 20, 2020, 6:23 PM
 */
#define FOSC (32000000)
#define FCY (FOSC/2)

//#include <p24FJ256DA210.h>
#include "xc.h"
#include "libpic30.h"
#include "structs.h"
#include "initialize.h"
#include "balancing.h"
#include "readsymux.h"

//#define FOSC (32000000)
//#define FCY (FOSC/2)
//#include "auxiliar.h"

////////////////////////////////////
// Comunicaciones por USB
////////////////////////////////////
char RX(void);
int TX (char datum);

int USB_comm (void)
{
    char RX_data;
    int error;
    Tfeedback datos;

```

```
Tfeedback Response(char RX_data);
```

```
    RX_data = RX();                // Verificar si ha llegado un dato
    if (RX_data != 'X')            // Se recibe el dato correctamente
    {
        datos = Response(RX_data);

        if (datos.feedback_fault == 0)
            // Si se recibe el return de I2C se envían los datos solicitados
            {
                error = TX(datos.integer);    // Envía la parte entera
                if (error == 1)                // Si no se envía, espera y se vuelve a intentar
                {
                    __delay_us(5);
                    TX (datos.integer);
                }
                error = TX(datos.decimal);    // Envía la parte decimal
                if (error == 1)                // Si no se envía, espera y se vuelve a intentar
                {
                    __delay_us(5);
                    TX (datos.decimal);
                }
            }
    }
    return datos.feedback_fault;
}
```

```
////////////////////////////////////
// Configuración de los UART del PIC
////////////////////////////////////
```

```
void config_UART (void)
```

```

{
// Configuración de los registros
U1MODEbits.UARTEN = 1;           // Se habilitan los puertos UART
U1MODEbits.USIDL = 0;           // Se continua la operación en Idle mode
U1MODEbits.IREN = 0;           // Se deshabilita encoder y decoder IrDA
U1MODEbits.RTSMD = 0;          // U1RTS = 0 is in Flow Control Mode
U1MODEbits.UEN = 0b00;
U1MODEbits.WAKE = 0;           // Se deshabilita el Wake-up
U1MODEbits.LPBACK = 0;         // Se deshabilita el Loopback Mode
U1MODEbits.ABAUD = 0;          // Se deshabilita la medición de los baudios
U1MODEbits.RXINV = 0;          // U1RX Idle State is 1
U1MODEbits.BRGH = 0;           // Modo de velocidad estándar
U1MODEbits.PDSEL = 0;          // Dato de 8-bit
U1MODEbits.STSEL = 0;          // Se habilita el STOP cuando está a 1

// Estado y control de los registros
U1STAbits.UTXISEL0 = 0;        // Modo de interrupción de transmisión
U1STAbits.UTXISEL1 = 0;        // Modo de interrupción de transmisión
U1STAbits.UTXINV = 0;          // U1TX Idle State = 1
U1STAbits.UTXBRK = 0; // Sync Break Transmission está deshabilitada o
completa
U1STAbits.UTXEN = 1;           // Se habilita el transmisor de las UART
U1STAbits.URXISEL = 0b00;
// 0x Se produce interrupción después de que un carácter sea recibido por RX
U1STAbits.ADDEN = 0;           // Se deshabilita el Address Detect Mode

// Velocidad Baudios
U1BRG = 51;                     // 19230 Baudios con 32 MHz
}

////////////////////////////////////
// Configuración de la recepción de datos (RX)
////////////////////////////////////

```

```

char RX (void)
{
    char RX_data;

    if (U1STAbits.FERR == 1) {}           // Se detecta Framing error
    if (U1STAbits.OERR == 1)             // Desbordamiento en la recepción
    {
        U1STAbits.OERR =0;               // Reseteo del error para seguir recibiendo
    }
    if (U1STAbits.URXDA == 1)
    {
        RX_data = U1RXREG;               // Se reciben datos
    }
    else RX_data = 'X';                  // No hay dato recibido
    return RX_data;
}

```

```

////////////////////////////////////
// Configuración de la transmisión de datos (TX)
////////////////////////////////////

```

```

int TX (char datum)
{
    int error;

    error = 0;
    if (U1STAbits.UTXBF == 1)
    {
        error = 1;    // Si el buffer de transmisión está desbordado, da error
    }
    else U1TXREG = datum;
}

```

```

return error;
}

////////////////////////////////////
// Estados del dato de respuesta
////////////////////////////////////

Tfeedback Response (char RX_data)
{
int integer, decimal, aux;
Tfeedback datos;
Tbattery battery_info;

datos.feedback_fault = 0;

switch (RX_data) // Casos para comprobar qué se pide
{
case 'a': // Celda 0
integer = divide_get_voltage_cells_pose(100,0);
decimal = subtraction_get_voltage_cells_pose((integer*100),0);
break;

case 'b': // Celda 1
integer = divide_get_voltage_cells_pose(100,1);
decimal = subtraction_get_voltage_cells_pose((integer*100),1);
break;

case 'c': // Celda 2
integer = divide_get_voltage_cells_pose(100,2);
decimal = subtraction_get_voltage_cells_pose((integer*100),2);
break;
}
}

```

```
case 'd':                // Celda 3
    integer = divide_get_voltage_cells_pose(100,3);
    decimal = subtraction_get_voltage_cells_pose((integer*100),3);
break;

case 'e':                // Celda 4
    integer = divide_get_voltage_cells_pose(100,4);
    decimal = subtraction_get_voltage_cells_pose((integer*100),4);
break;

case 'f':                // Celda 5
    integer = divide_get_voltage_cells_pose(100,5);
    decimal = subtraction_get_voltage_cells_pose((integer*100),5);
break;

case 'g':                // Celda 6
    integer = divide_get_voltage_cells_pose(100,6);
    decimal = subtraction_get_voltage_cells_pose((integer*100),6);

case 'h':                // Celda 7
    integer = divide_get_voltage_cells_pose(100,7);
    decimal = subtraction_get_voltage_cells_pose((integer*100),7);
break;

case 'i':                // Celda 8
    integer = divide_get_voltage_cells_pose(100,8);
    decimal = subtraction_get_voltage_cells_pose((integer*100),8);
break;

case 'j':                // Celda 9
    integer = divide_get_voltage_cells_pose(100,9);
    decimal = subtraction_get_voltage_cells_pose((integer*100),9);
break;
```

```

case 'k':                // Celda 10
    integer = divide_get_voltage_cells_pose(100,10);
    decimal = subtraction_get_voltage_cells_pose((integer*100),10);
break;

case 'l':                // Celda 11
    integer = divide_get_voltage_cells_pose(100,11);
    decimal = subtraction_get_voltage_cells_pose((integer*100),11);
break;

case 'm':                // Celda 12
    integer = divide_get_voltage_cells_pose(100,12);
    decimal = subtraction_get_voltage_cells_pose((integer*100),12);
break;

case 'n':                // Celda 13
    integer = divide_get_voltage_cells_pose(100,13);
    decimal = subtraction_get_voltage_cells_pose((integer*100),13);
break;

case 'o':                // Celda 14
    integer = divide_get_voltage_cells_pose(100,14);
    decimal = subtraction_get_voltage_cells_pose((integer*100),14);
break;

case 'p':                // Celda 15
    integer = divide_get_voltage_cells_pose(100,15);
    decimal = subtraction_get_voltage_cells_pose((integer*100),15);
break;

case 'l':                // Corriente
    aux = get_Current_value()/10;

```

```

integer = aux/100;
decimal = aux - (integer*100);
break;

case 'C':                // Calibrado
integer = get_calib_state_pose(1);
decimal = get_calib_state_pose(0);
break;

case 'R':                // Sensor Temperatura 1
if (get_Temper1_value() > 9000)
{
integer = 99;
decimal = 99;
}
else
{
integer = (int) get_Temper1_value();
decimal = (int) (get_Temper1_value() - integer)*100;
}
break;

case 'S':                // Sensor Temperatura 2
if (get_Temper2_value() > 9000)
{
integer = 99;
decimal = 99;
}
else
{
integer = (int) get_Temper2_value();
decimal = (int) (get_Temper2_value() - integer)*100;
}

```

```
break;
```

```
case 'T':                // Sensor Temperatura 3
    if (get_Temper3_value() > 9000)
    {
        integer = 99;
        decimal = 99;
    }
    else
    {
        integer = (int) get_Temper3_value();
        decimal = (int) (get_Temper3_value() - integer)*100;
    }
break;
```

```
case 'V':                // Sensor Temperatura 4
    if (get_Temper4_value() > 9000)
    {
        integer = 99;
        decimal = 99;
    }
    else
    {
        integer = (int) get_Temper4_value();
        decimal = (int) (get_Temper4_value() - integer)*100;
    }
break;
```

```
case 'A':                // Tensión nominal de las celdas
    integer = battery_info.SOA.rate_tension/100;
    decimal = battery_info.SOA.rate_tension - (integer*100);
break;
```

```

case 'B':                // Tensión máxima de carga de las celdas
    integer = battery_info.SOA.max_chg_tension/100;
    decimal = battery_info.SOA.max_chg_tension - (integer*100);
break;

case 'D':                // Tensión máxima de descarga de las celdas
    integer = battery_info.SOA.max_dchg_tension/100;
    decimal = battery_info.SOA.max_dchg_tension - (integer*100);
break;

case 'E':                // Tensión de seguridad de las celdas
    integer = battery_info.SOA.sec_tension/100;
    decimal = battery_info.SOA.sec_tension - (integer*100);
break;

case 'F':                // Temperatura máxima de las celdas
    integer = (int) battery_info.SOA.t_max;
    decimal = (int) ((battery_info.SOA.t_max - integer)*100);
break;

case 'G':                // Número de celdas usadas
    integer = battery_info.n_used_blockB_cells;
    decimal = battery_info.n_used_blockA_cells;
break;

case 'H':                // Corriente máxima de carga
    aux = battery_info.max_chg_current/10;
    integer = aux/100;
    decimal = aux - (integer*100);
break;

case 'J':                // Corriente mínima de carga
    aux = battery_info.min_chg_current/10;

```

```

integer = aux/100;
decimal = aux - (integer*100);
break;

case 'K':                // Corriente de pico de carga
    aux = battery_info.peak_chg_current/10;
    integer = aux/100;
    decimal = aux - (integer*100);
break;

case 'L':                // Corriente máxima de descarga
    aux = battery_info.max_dchg_current/10;
    integer = aux/100;
    decimal = aux - (integer*100);
break;

case 'M':                // Corriente de pico de descarga
    aux = battery_info.peak_dchg_current/10;
    integer = aux/100;
    decimal = aux - (integer*100);
break;

default:
    datos.feedback_fault = 1;
    //Alguna parte del código incorrecta y no se ajusta a ningún caso
}
datos.integer = integer;
datos.decimal = decimal;
return datos;
}

```

```

/*
 * File: I2C.c
 * Author: Ahullonet
 *
 * Created on July 22, 2020, 3:58 PM
 */

#include <xc.h>
#include "I2C.h"
#include "structs.h"
#include "balancing.h"

////////////////////////////////////
// Configuración del I2C
////////////////////////////////////

Tfeedback Response(char RX_data);

void I2C_init (unsigned char F)
{
    int temp;

    I2C2BRG = F;          // Velocidad de la com I2C
    I2C2CONbits.I2CEN = 0; // Deshabilita módulo I2C2
    I2C2CONbits.DISSLW = 1; // Deshabilita el slew rate control
    IFS3bits.SI2C2IF = 0; // Clear Slave Interrupt Flag
    IFS3bits.MI2C2IF = 0; // Clear Master Interrupt Flag
    I2C2CONbits.IPMIEN = 1; // Módulo I2C operando como esclavo
    I2C2CONbits.I2CEN = 1; // Se habilita el I2C2 y se configuran los pines
    temp = I2C2RCV;      // Se lee el buffer y se aclara
    IEC3bits.MI2C2IE = 0; // Se deshabilita la interrupción del módulo master
    IEC3bits.SI2C2IE = 0; // Se deshabilita la interrupción del slave
    I2C2CONbits.SCLREL = 0; // Se mantiene SCLx clock low
    I2C2CONbits.A10M = 0; // I2CxADD es un dirección esclava de 7 bits

```

```

    I2C2ADD = SLAVE_ADDRESS; // Direccion del esclavo
}
////////////////////////////////////
// Comunicaciones por I2C
////////////////////////////////////

int I2C_comm (void)
{
    char RX_data;
    Tfeedback data;

    if (I2C2STATbits.RBF == 1) {
        RX_data = I2C2RCV;

        if (RX_data == SLAVE_ADDRESS) // Si el dato es la dirección del BMS
        {
            AckI2C();           //Se genera la señal de acknowledgement
            RX_data = SlaveReadI2C(); //Espera a la lectura del registro
            AckI2C();           //Genera señal de ACK
            data = Response(RX_data); //Prepara los datos solicitados
            RX_data = SlaveReadI2C(); //Espera a la lectura
            AckI2C();           //Genera señal de ACK

            if ((RX_data == SLAVE_ADDRESS) && (data.feedback_fault==0)) {
                //Si la petición es correcta se envían los datos
                SlaveWriteI2C(data.integer); //Parte entera
                SlaveWriteI2C(data.decimal); //Parte decimal
            }
        }
    }
    return data.feedback_fault;
}

```

```

////////////////////////////////////
// Lectura del esclavo en I2C
////////////////////////////////////

unsigned char SlaveReadI2C(void)
{
    while(!I2C2STATbits.RBF);
    I2C2STATbits.I2COV = 0;
    return(I2C2RCV);
}

```

```

////////////////////////////////////
//Escritura del esclavo en I2C
////////////////////////////////////

```

```

void SlaveWriteI2C (unsigned char data_out)
{
    I2C2TRN = data_out;    //Dato transferido al registro I2C2TRN
    I2C2CONbits.SCLREL = 1;    //Liberar el reloj
}

```

```

////////////////////////////////////
// Señal de acknowledgement
////////////////////////////////////

```

```

void AckI2C(void)
{
    I2C2CONbits.ACKDT = 0;
    I2C2CONbits.ACKEN = 1;
}

```

```

/*
 * File: I2C.h
 * Author: Ahullonet
 *
 * Created on August 19, 2020, 7:21 PM
 */

#ifndef I2C_H
#define I2C_H

#define F_100KHz 157 // I2C Baudrate Normal Mode
#define F_394KHz 38 // I2C Baudrate Fast Mode
#define F_1MHz 13 // I2C Baudrate Ultra fast Mode
#define SLAVE_ADDRESS 0x28 // Address of the BMS

////////////////////
//Declaración de las funciones
////////////////////

void I2C_init(unsigned char);
void SlaveWriteI2C(unsigned char data_out);
unsigned char SlaveReadI2C(void);
int I2C_comm(void);
void AckI2C(void);

#endif

```

```

* File: structs.h
* Author: Ahullonet
* Created on September 11, 2020, 6:16 PM
#endif STRUCTS_H
#define    STRUCTS_H

typedef struct cell {
    int rate_tension;
    int max_chg_tension;
    int max_dchg_tension;
    int sec_tension;
    double t_max;
} Tcell;

typedef struct battery {
    int n_used_blockA_cells;    // Número de celdas conectadas en el bloque A
    int n_used_blockB_cells;    // Número de celdas conectadas en el bloque B
    long int max_chg_current;    // Corriente máxima de carga
    long int min_chg_current;    // Corriente mínima de carga
    long int peak_chg_current;   // Corriente máxima de pico de carga
    long int max_dchg_current;   // Corriente máxima de descarga
    long int peak_dchg_current;  // Corriente de pico máxima de descarga
    double chg_balancing;       // Porcentaje de balanceo en carga
    double stb_balancing;       // Porcentaje de balanceo durante estabilización
    Tcell SOA;                  // Datos SOA de la batería
} Tbattery;

typedef struct feedback {
    int integer;
    int decimal;
    char feedback_fault;
} Tfeedback;

#endif    /* STRUCTS_H */

```

```

/*
 * File:  thermos.c
 * Author: Ahullonet
 *
 * Created on July 21, 2020, 4:36 PM
 */
#include <p24FJ256DA210.h>
#define FOSC (32000000)
#define FCY (FOSC/2)

#include <xc.h>
#include <libpic30.h>
#include "thermos.h"
#include "structs.h"
#include "initialize.h"
#include "balancing.h"
#include "auxiliar.h"

#define HIGH 1
#define LOW 0
#define SET 1
#define RESET 0

////////////////////////////////////
// Sensor de Temperatura 1
////////////////////////////////////

#define T1_direc TRISBbits.TRISB5
#define T1_write LATBbits.LATB5
#define T1_read PORTBbits.RB5
#define T1_open_drain ODCBbits.ODB5 = 1 // Open drain = 1

```

```

////////////////////////////////////
// Sensor de Temperatura 2
////////////////////////////////////

#define T2_direc TRISBbits.TRISB4
#define T2_write LATBbits.LATB4
#define T2_read PORTBbits.RB4
#define T2_open_drain ODCBbits.ODB4 = 1 // Open drain = 1

////////////////////////////////////
// Sensor de Temperatura 3
////////////////////////////////////

#define T3_direc TRISBbits.TRISB3
#define T3_write LATBbits.LATB3
#define T3_read PORTBbits.RB3
#define T3_open_drain ODCBbits.ODB3 = 1 // Open drain = 1

////////////////////////////////////
// Sensor de Temperatura 4
////////////////////////////////////

#define T4_direc TRISBbits.TRISB2
#define T4_write LATBbits.LATB2
#define T4_read PORTBbits.RB2
#define T4_open_drain ODCBbits.ODB2 = 1 // Open drain = 1

double T1, T2, T3, T4;

////////////////////////////////////
// Configuración de los sensores como SALIDAS a LOW

```

////////////////////////////////////

```
void low_Thermos_drive(unsigned int T)
{
  switch (T)
  {
    case 1:
      T1_direc = OUTPUT;
      T1_open_drain;
      T1_write = LOW;
      break;

    case 2:
      T2_direc = OUTPUT;
      T2_open_drain;
      T2_write = LOW;
      break;

    case 3:
      T3_direc = OUTPUT;
      T3_open_drain;
      T3_write = LOW;
      break;

    case 4:
      T4_direc = OUTPUT;
      T4_open_drain;
      T4_write = LOW;
      break;
  }
}
```

////////////////////////////////////

```
// Configuración de los sensores como SALIDAS a HIGH
```

```
////////////////////////////////////
```

```
void high_Thermos_drive(unsigned int T)
```

```
{
```

```
  switch (T)
```

```
  {
```

```
    case 1:
```

```
      T1_direc = OUTPUT;
```

```
      T1_open_drain;
```

```
      T1_write = HIGH;
```

```
    break;
```

```
    case 2:
```

```
      T2_direc = OUTPUT;
```

```
      T2_open_drain;
```

```
      T2_write = HIGH;
```

```
    break;
```

```
    case 3:
```

```
      T3_direc = OUTPUT;
```

```
      T3_open_drain;
```

```
      T3_write = HIGH;
```

```
    break;
```

```
    case 4:
```

```
      T4_direc = OUTPUT;
```

```
      T4_open_drain;
```

```
      T4_write = HIGH;
```

```
    break;
```

```
  }
```

```
}
```

```
////////////////////////////////////  
// Lectura de los estados de los pines, para devolver valores  
// Se configuran los pines como entradas  
////////////////////////////////////
```

```
unsigned int read_Thermos(unsigned int T)  
{  
    unsigned int dat = 3;  
    switch (T)  
    {  
        case 1:  
            T1_direc = INPUT;  
            dat = T1_read;  
            break;  
  
        case 2:  
            T2_direc = INPUT;  
            dat = T2_read;  
            break;  
  
        case 3:  
            T3_direc = INPUT;  
            dat = T3_read;  
            break;  
  
        case 4:  
            T4_direc = INPUT;  
            dat = T4_read;  
            break;  
    }  
    return dat;  
}
```

```

////////////////////////////////////
// Secuencia de reseteo
// Se devuelve el pulso de frecuencia desde el esclavo
////////////////////////////////////

```

```

unsigned int reset_Thermos(unsigned int T)
{
    unsigned int presence;
    low_Thermos_drive(T);
    __delay_us(480);      // Todas las salidas a low
    high_Thermos_drive(T);
    __delay_us(70);      // Libera el bus
    presence = read_Thermos(T);
    __delay_us(410);     // Detección de presencia
    high_Thermos_drive(T); // Libera el bus

    return presence;
}

```

```

////////////////////////////////////
// Función de escritura de un bit
////////////////////////////////////

```

```

void write_bit_Thermos (unsigned int write_bit, unsigned int T)
{
    if (write_bit)
    {
        low_Thermos_drive(T);
        __delay_us(6);      // Pasar todo a LOW
        high_Thermos_drive(T);
        __delay_us(64);     // Se escribe el bit '1'
    }
    else

```

```

{
  low_Thermos_drive(T);
  __delay_us(60);          // Pasar todo a low
  high_Thermos_drive(T);
  __delay_us(10);         // Escribir un 0
}
}

////////////////////////////////////
// Función de lectura de un bit desde el esclavo
////////////////////////////////////

unsigned int read_bit_Thermos (unsigned int T)
{
  unsigned int dato;
  low_Thermos_drive(T);
  __delay_us(6);          // Pasar todo a low
  high_Thermos_drive(T);
  __delay_us(9);          // Liberar el bus
  dato = read_Thermos(T);
  __delay_us(55);         // Lee el estado de los pines de los Thermos

  return dato;
}

////////////////////////////////////
// Función de escritura de un BYTE al esclavo
////////////////////////////////////

void write_byte_Thermos (int dato, unsigned int T)
{
  int i;
  for (i=0;i<0;i++)

```

```

    {
        write_bit_Thermos(dato & 0x01, T); // Se envía el primer LS-bit
        dato >>= 1; // Se aumenta la dirección en 1 para el siguiente bit del byte
    }
}

```

```

////////////////////////////////////

```

```

// Función de lectura del byte

```

```

////////////////////////////////////

```

```

int read_byte_Thermos (unsigned int T)

```

```

{
    int i, dato = 0;
    for (i=0;i<8;i++)
    {
        dato >>= 1; // Se aumenta la dirección en 1 para el siguiente bit del byte
        if (read_bit_Thermos(T));
        {
            dato |= 0x80; // Si es 1, se setea el MS-bit
        }
    }
    return dato;
}

```

```

////////////////////////////////////

```

```

// Inicialización de los sensores

```

```

////////////////////////////////////

```

```

void init_Thermos (void)

```

```

{
    T1 = 0.0;
    T2 = 0.0;
    T3 = 0.0;
}

```

```

T4 = 0.0;
if (reset_Thermos(1) == 0) T1 = 1;    // Si se detecta el sensor 1
if (reset_Thermos(2) == 0) T2 = 1;    // Si se detecta el sensor 2
if (reset_Thermos(3) == 0) T3 = 1;    // Si se detecta el sensor 3
if (reset_Thermos(4) == 0) T4 = 1;    // Si se detecta el sensor 4

if (T1)
{
    write_byte_Thermos(0xCC, 1);        // Comando Skip_ROM
    write_byte_Thermos(0x44, 1);        // Comando Conver-T
}
if (T2)
{
    write_byte_Thermos(0xCC, 2);        // Comando Skip_ROM
    write_byte_Thermos(0x44, 2);        // Comando Conver-T
}
if (T3)
{
    write_byte_Thermos(0xCC, 3);        // Comando Skip_ROM
    write_byte_Thermos(0x44, 3);        // Comando Conver-T
}
if (T4)
{
    write_byte_Thermos(0xCC, 4);        // Comando Skip_ROM
    write_byte_Thermos(0x44, 4);        // Comando Conver-T
}
}

////////////////////////////////////
// Lectura de la temperatura
////////////////////////////////////

void read_temp_Thermos (void)

```

```

{

int Temp1, Temp2, Temp3, Temp4;
int TempL1, TempH1, TempL2, TempH2, TempL3, TempH3, TempL4, TempH4;

if (T1)
{
  reset_Thermos(1);
  write_byte_Thermos(0xCC, 1);          // Comando Skip ROM
  write_byte_Thermos(0xBE, 1);          // Comando Read_Scratchpad
  TempL1 = read_byte_Thermos(1);        // Lee el byte LSB
  TempH1 = read_byte_Thermos(1);        // Lee el byte MSB
  Temp1 = (TempH1 << 8) | TempL1; // Desplaza 8 bits izq a MSB y OR a la LSB
  set_Temper1_value(Temp1 * 0.0625);    // Offset incorrecto para el mío
}
else                                     // No detecta el sensor
{
  set_Temper1_value(9999);
}
if (T2)
{
  reset_Thermos(2);
  write_byte_Thermos(0xCC, 2);          // Comando Skip ROM
  write_byte_Thermos(0xBE, 2);          // Comando Read_Scratchpad
  TempL2 = read_byte_Thermos(2);        // Lee el byte LSB
  TempH2 = read_byte_Thermos(2);        // Lee el byte MSB
  Temp2 = (TempH2 << 8) | TempL2; // Desplaza 8 bits izq a MSB y OR a la LSB
  set_Temper2_value(Temp2 * 0.0625);    //Offset incorrecto para el mío
}
else                                     // No detecta el sensor
{
  set_Temper2_value(9999);
}
}

```

```

if (T3)
{
  reset_Thermos(3);
  write_byte_Thermos(0xCC, 3);          // Comando Skip ROM
  write_byte_Thermos(0xBE, 3);         // Comando Read_Scratchpad
  TempL3 = read_byte_Thermos(3);       // Lee el byte LSB
  TempH3 = read_byte_Thermos(3);       // Lee el byte MSB
  Temp3 = (TempH3 << 8) | TempL3;      // Desplaza 8 bits izq a MSB y OR a la LSB
  set_Temper3_value(Temp3 * 0.0625);   //Offset incorrecto para el mío
}
else                                  // No detecta el sensor
{
  set_Temper3_value(9999);
}
if (T4)
{
  reset_Thermos(4);
  write_byte_Thermos(0xCC, 4);         // Comando Skip ROM
  write_byte_Thermos(0xBE, 4);         // Comando Read_Scratchpad
  TempL4 = read_byte_Thermos(4);       // Lee el byte LSB
  TempH4 = read_byte_Thermos(4);       // Lee el byte MSB
  Temp4 = (TempH4 << 8) | TempL4;      // Desplaza 8 bits izq a MSB y OR a la LSB
  set_Temper4_value(Temp4 * 0.0625);   //Offset incorrecto para el mío
}
else                                  // No detecta el sensor
{
  set_Temper4_value(9999);
}
}

```

```

/*
* File:  thermos.h
* Author: Ahullonet
* Created on September 11, 2020, 6:20 PM
#endif THERMOS_H
#define    THERMOS_H

#define N_TOTAL_CELLS 16    // Máximo de 16 celdas
#define N_BLOCK_CELLS 8    // Máximo de 8 celdas por bloque
#define T_MAX_PEAK 100    // Tiempo máx de pico
#define T_FAN 50.0        // Por encima de 50° se enciende
#define POSITIVE_SENS_A 500 // Por encima empieza la descarga
#define POSITIVE_SENS_B 350 // Por debajo termina la descarga
#define NEGATIVE_SENS_A -500 // Por debajo empieza la carga
#define NEGATIVE_SENS_B -350 // Por arriba termina la descarga

// TRIS
#define OUTPUT 0
#define INPUT 1

// ANS
#define ANALOG 1
#define DIGITAL 0

// ODC
#define OPEN_DRAIN 1
#define AVERAGE 50    // Lecturas necesarias para hacer la media

// #define SW_OFF PORTBbits.RB8 // Switch OFF = 1 y ON = 0
#define CHG    LATEbits.LATE6 // Carga
#define DCHG   LATEbits.LATE5 // Descarga
#define FAN    LATEbits.LATE7 // Ventilador
#define EMER_LED LATGbits.LATG6 // LED de emergencia
#endif    /* THERMOS_H */

```

ANEXO 4: SIMULACIÓN DEL DISPOSITIVO



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

SIMULACIÓN NÚMERO 1:

Simulación real en la que se conecta una carga que consume 10Ah, las celdas presentan voltaje nominal y todos los sensores y parámetros medidos se encuentran dentro de los límites. El resultado es el establecimiento del “Modo de DESCARGA” sin necesidad de calibración por parte de las celdas de ninguno de los dos bloques. Este estado activa el optoacoplador de descarga mientras sigue midiendo y verificando los límites de los sensores hasta que la batería cambie de estado.

{

El programa ha empezado porque Init es 1

Se ha configurado el oscilador principal

Se han configurado los puertos conectados al microcontrolador

Se han configurado los puertos y dispositivos serie

Se ha configurado la comunicacion I2C

Ventilador deshabilitado

LED de emergencia apagado

La variable MA es 1

La variable CHARGED = 1 indica que la batería no está cargada o no ha habido carga previa

SE ESTABLECEN LOS PARAMETROS DE LA BATERIA A GESTIONAR

Se determina que el número de celdas en el bloque A es 8

Se determina que el número de celdas en el bloque B es 8

Se determina la corriente máxima de carga en -12500

Se determina la corriente mínima de carga en -500

Se determina la corriente de pico de carga en -25000

Se determina la corriente máxima de descarga en 35000

Se determina el pico de corriente de carga en -25000

Se determina el rango de desequilibrio en la carga en 1.007000

Se determina el rango de desequilibrio una vez cargado en 1.005000

Número de celdas conectadas = 16

Se determina la temperatura máxima de funcionamiento en 55.000000

Se determina la tensión máxima de funcionamiento en 3650

Se determina la tensión máxima de carga en 3600

Se determina la tension nominal de la celda en 3200
Se determina la tension maxima de descarga en 55.000000
Se establece la temperatura ambiente en los sensores de temperatura, es 25.000000
La calibracion del bloque A se deshabilita, igual a 0
La calibracion del bloque B se deshabilita, igual a 0
Se habilitan los canales 1, 2 y 3 del ADC interno
Corriente = 10000
Voltaje de la celda 1 = 3200
Voltaje de la celda 9 = 3200
Corriente = 10000
Voltaje de la celda 2 = 3200
Voltaje de la celda 10 = 3200
Corriente = 10000
Voltaje de la celda 3 = 3200
Voltaje de la celda 11 = 3200
Corriente = 10000
Voltaje de la celda 4 = 3200
Voltaje de la celda 12 = 3200
Corriente = 10000
Voltaje de la celda 5 = 3200
Voltaje de la celda 13 = 3200
Corriente = 10000
Voltaje de la celda 6 = 3200
Voltaje de la celda 14 = 3200
Corriente = 10000
Voltaje de la celda 7 = 3200
Voltaje de la celda 15 = 3200
Tension total de la bateria = 51200
El programa ha inicializado!
SW_ON = 1
Equipo maestro encendido
Sensores de temperatura inicializados

Corriente = 10000

Voltaje de la celda 1 = 3200

Voltaje de la celda 9 = 3200

Corriente = 10000

Voltaje de la celda 2 = 3200

Voltaje de la celda 10 = 3200

Corriente = 10000

Voltaje de la celda 3 = 3200

Voltaje de la celda 11 = 3200

Corriente = 10000

Voltaje de la celda 4 = 3200

Voltaje de la celda 12 = 3200

Corriente = 10000

Voltaje de la celda 5 = 3200

Voltaje de la celda 13 = 3200

Corriente = 10000

Voltaje de la celda 6 = 3200

Voltaje de la celda 14 = 3200

Corriente = 10000

Voltaje de la celda 7 = 3200

Voltaje de la celda 15 = 3200

Tension total de la bateria = 51200

Datos del ADC leidos

Se procede a la verificacion de los limites de los sensores

Sensores de temperatura:

Primer sensor (ambiente) conectado y midiendo 25.000000

Por tanto, ventilador igual a 0

Error T2 = 0

Error T3 = 0

Error T4 = 0

No hay fallo de temperatura, se continua verificando. ErrorT = 0

Voltaje celda 1 = 3200

Voltaje celda 9 = 3200

Voltaje celda 2 = 3200

Voltaje celda 10 = 3200

Voltaje celda 3 = 3200

Voltaje celda 11 = 3200

Voltaje celda 4 = 3200

Voltaje celda 12 = 3200

Voltaje celda 5 = 3200

Voltaje celda 13 = 3200

Voltaje celda 6 = 3200

Voltaje celda 14 = 3200

Voltaje celda 7 = 3200

Voltaje celda 15 = 3200

NO hay error de voltaje, se sigue verificando

La corriente instantanea es = 10000

La corriente esta dentro de los limites

Sin errores en las lecturas de voltaje de descarga de la bateria, se sigue verificando

Se comprueba que la tension de la bateria esta dentro de los limites

Tension de bateria medida = 51200 < 57600 Tension maxima carga

Verificacion completa y sin errores

Error final de verificacion = 0

La calibracion del bloque A se deshabilita, igual a 0

La calibracion del bloque B se deshabilita, igual a 0

MODO DESCARGA

Entramos a MOSFET_enable!

La variable DCHG es 1

La variable CHG es 0

Por tanto, solo se permite la descarga

Equipo maestro encendido

Sensores de temperatura inicializados

Corriente = 10000

Voltaje de la celda 1 = 3200

Voltaje de la celda 9 = 3200

Corriente = 10000
Voltaje de la celda 2 = 3200
Voltaje de la celda 10 = 3200
Corriente = 10000
Voltaje de la celda 3 = 3200
Voltaje de la celda 11 = 3200
Corriente = 10000
Voltaje de la celda 4 = 3200
Voltaje de la celda 12 = 3200
Corriente = 10000
Voltaje de la celda 5 = 3200
Voltaje de la celda 13 = 3200
Corriente = 10000
Voltaje de la celda 6 = 3200
Voltaje de la celda 14 = 3200
Corriente = 10000
Voltaje de la celda 7 = 3200
Voltaje de la celda 15 = 3200
Tension total de la bateria = 51200

Datos del ADC leidos

Se procede a la verificacion de los limites de los sensores

Sensores de temperatura:

Primer sensor (ambiente) conectado y midiendo 25.000000

Por tanto, ventilador igual a 0

Error T2 = 0

Error T3 = 0

Error T4 = 0

No hay fallo de temperatura, se continua verificando. ErrorT = 0

Voltaje celda 1 = 3200

Voltaje celda 9 = 3200

Voltaje celda 2 = 3200

Voltaje celda 10 = 3200

Voltaje celda 3 = 3200

Voltaje celda 11 = 3200

Voltaje celda 4 = 3200

Voltaje celda 12 = 3200

Voltaje celda 5 = 3200

Voltaje celda 13 = 3200

Voltaje celda 6 = 3200

Voltaje celda 14 = 3200

Voltaje celda 7 = 3200

Voltaje celda 15 = 3200

NO hay error de voltaje, se sigue verificando

La corriente instantanea es = 10000

La corriente esta dentro de los limites

Sin errores en las lecturas de voltaje de descarga de la bateria, se sigue verificando

Se comprueba que la tension de la bateria esta dentro de los limites

Tension de bateria medida = 51200 < 57600 Tension maxima carga

Verificacion completa y sin errores

Error final de verificacion = 0

La calibracion del bloque A se deshabilita, igual a 0

La calibracion del bloque B se deshabilita, igual a 0

MODO DESCARGA

Entramos a MOSFET_enable!

La variable DCHG es 1

La variable CHG es 0

Por tanto, solo se permite la descarga

}

FINALIZACIÓN DE LA SIMULACIÓN 1

SIMULACIÓN NÚMERO 2:

Simulación en la que la batería se encuentra en el “Modo de CARGA” a 600W (10,416A a 57,6V) en el momento en que el bloque B de celdas rebasa el límite de voltaje máximo, el BMS lo diagnostica y deja de cargar la batería, activándose el LED de emergencia y el “Modo de DESCARGA”.

{

El programa ha empezado porque Inıt es 1

Se ha configurado el oscilador principal

Se han configurado los puertos conectados al microcontrolador

Se han configurado los puertos y dispositivos serie

Se ha configurado la comunicacion I2C

Entramos a MOSFET_enable!

La variable DCHG es 0

La variable CHG es 1

Por tanto, solo se permite la carga

Ventilador deshabilitado

LED de emergencia apagado

La variable MA es 1

La variable CHARGED = 1 indica que la batería no se ha descargado despues de cargar o que todavia carga

SE ESTABLECEN LOS PARAMETROS DE LA BATERIA A GESTIONAR

Se determina que el numero de celdas en el bloque A es 8

Se determina que el numero de celdas en el bloque B es 8

Se determina la corriente maxima de carga en -12500

Se determina la corriente minima de carga en -500

Se determina la corriente de pico de carga en -25000

Se determina la corriente maxima de descarga en 35000

Se determina el pico de corriente de carga en -25000

Se determina el rango de desequilibrio en la carga en 1.007000

Se determina el rango de desequilibrio una vez cargado en 1.005000

Numero de celdas conectadas = 16

Se determina la temperatura maxima de funcionamiento en 55.000000

Se determina la tension maxima de funcionamiento en 3650

Se determina la tension maxima de carga en 3600
Se determina la tension nominal de la celda en 3200
Se determina la tension maxima de descarga en 55.000000
Se establece la temperatura ambiente en los sensores de temperatura, es 25.000000
La calibracion del bloque A se deshabilita, igual a 0
La calibracion del bloque B se deshabilita, igual a 0
Se habilitan los canales 1, 2 y 3 del ADC interno
Corriente = -10416
Tension total de la bateria = 58096
El programa ha inicializado!
SW_ON = 1
Equipo maestro encendido
Sensores de temperatura inicializados
Corriente = -10416
Tension total de la bateria = 58096
Datos del ADC leidos
Se procede a la verificacion de los limites de los sensores
Sensores de temperatura:
Primer sensor (ambiente) conectado y midiendo 25.000000
Por tanto, ventilador igual a 0
Error T2 = 0
Error T3 = 0
Error T4 = 0
No hay fallo de temperatura, se continua verificando. ErrorT = 0
Voltaje celda 1 = 3592
Voltaje celda 9 = 3670
La celda 1 del bloque B esta por encima del voltaje maximo
Error = 3
DCHG_CHG = 1. Solamente permite la descarga
CHARGED = 3. Bateria cargada pero con una celda sobrecargada
Voltaje celda 2 = 3592
Voltaje celda 10 = 3670

La celda 2 del bloque B esta por encima del voltaje maximo

Error = 3

DCHG_CHG = 1. Solamente permite la descarga

CHARGED = 3. Bateria cargada pero con una celda sobrecargada

Voltaje celda 3 = 3592

Voltaje celda 11 = 3670

La celda 3 del bloque B esta por encima del voltaje maximo

Error = 3

DCHG_CHG = 1. Solamente permite la descarga

CHARGED = 3. Bateria cargada pero con una celda sobrecargada

Voltaje celda 4 = 3592

Voltaje celda 12 = 3670

La celda 4 del bloque B esta por encima del voltaje maximo

Error = 3

DCHG_CHG = 1. Solamente permite la descarga

CHARGED = 3. Bateria cargada pero con una celda sobrecargada

Voltaje celda 5 = 3592

Voltaje celda 13 = 3670

La celda 5 del bloque B esta por encima del voltaje maximo

Error = 3

DCHG_CHG = 1. Solamente permite la descarga

CHARGED = 3. Bateria cargada pero con una celda sobrecargada

Voltaje celda 6 = 3592

Voltaje celda 14 = 3670

La celda 6 del bloque B esta por encima del voltaje maximo

Error = 3

DCHG_CHG = 1. Solamente permite la descarga

CHARGED = 3. Bateria cargada pero con una celda sobrecargada

Voltaje celda 7 = 3592

Voltaje celda 15 = 3670

La celda 7 del bloque B esta por encima del voltaje maximo

Error = 3

DCHG_CHG = 1. Solamente permite la descarga

CHARGED = 3. Bateria cargada pero con una celda sobrecargada

Se apaga el MOSFET de carga

Como hay error de sobrevoltaje, SE HABILITA EL ESTADO DE EMERGENCIA

La corriente instantanea = 2756mA

La corriente esta dentro de los limites

Se comprueba que la tension de la bateria esta dentro de los limites

Tension de bateria medida = 58096 < 57600 Tension maxima carga?

Como ha habido algun error en la verificacion, se enciende el LED de emergencia

Error final de verificacion = 3

La calibracion del bloque A se deshabilita, igual a 0

La calibracion del bloque B se deshabilita, igual a 0

MODO DESCARGA

Entramos a MOSFET_enable!

La variable DCHG es 1

La variable CHG es 0

Por tanto, solo se permite la descarga

}

FINALIZACIÓN DE LA SIMULACIÓN 2

SIMULACIÓN NÚMERO 3:

Situación en la que se encuentra la batería en situación de carga a 600W (10,45A a 57,42V), dentro de los límites de carga pero con el bloque A de celdas desequilibrado del bloque B. Este estado activa el balanceo de las celdas con más voltaje, en este caso, todo el bloque A.

{

El programa ha empezado porque Inıt es 1

Se ha configurado el oscilador principal

Se han configurado los puertos conectados al microcontrolador

Se han configurado los puertos y dispositivos serie

Se ha configurado la comunicacion I2C

Entramos a MOSFET_enable!

La variable DCHG es 0

La variable CHG es 1

Por tanto, solo se permite la carga

Ventilador deshabilitado

LED de emergencia apagado

La variable MA es 1

La variable CHARGED = 1 indica que la batería no se ha descargado despues de cargar o que todavia carga

SE ESTABLECEN LOS PARAMETROS DE LA BATERIA A GESTIONAR

Se determina que el numero de celdas en el bloque A es 8

Se determina que el numero de celdas en el bloque B es 8

Se determina la corriente maxima de carga en -12500

Se determina la corriente minima de carga en -500

Se determina la corriente de pico de carga en -25000

Se determina la corriente maxima de descarga en 35000

Se determina el pico de corriente de carga en -25000

Se determina el rango de desequilibrio en la carga en 1.007000

Se determina el rango de desequilibrio una vez cargado en 1.005000

Numero de celdas conectadas = 16

Se determina la temperatura maxima de funcionamiento en 55.000000

Se determina la tension maxima de funcionamiento en 3650

Se determina la tension maxima de carga en 3600
Se determina la tension nominal de la celda en 3200
Se determina la tension maxima de descarga en 55.000000
Se establece la temperatura ambiente en los sensores de temperatura, es 25.000000
La calibracion del bloque A se deshabilita, igual a 0
La calibracion del bloque B se deshabilita, igual a 0
Se habilitan los canales 1, 2 y 3 del ADC interno
Corriente = -10416
Tension total de la bateria = 53096
El programa ha inicializado!
SW_ON = 1
Equipo maestro encendido
Sensores de temperatura inicializados
Corriente = -10416
Tension total de la bateria = 53096
Datos del ADC leidos
Se procede a la verificacion de los limites de los sensores
Sensores de temperatura:
Primer sensor (ambiente) conectado y midiendo 25.000000
Por tanto, ventilador igual a 0
Error T2 = 0
Error T3 = 0
Error T4 = 0
No hay fallo de temperatura, se continua verificando. ErrorT = 0
Voltaje celda 0 = 3336
Voltaje celda 8 = 3301
Voltaje celda 1 = 3336
Voltaje celda 9 = 3301
Voltaje celda 2 = 3336
Voltaje celda 10 = 3301
Voltaje celda 3 = 3336
Voltaje celda 11 = 3301

Voltaje celda 4 = 3336

Voltaje celda 12 = 3301

Voltaje celda 5 = 3336

Voltaje celda 13 = 3301

Voltaje celda 6 = 3336

Voltaje celda 14 = 3301

Voltaje celda 7 = 3336

Voltaje celda 15 = 3301

La corriente instantanea = -10416mA

La corriente esta dentro de los limites

Se comprueba que la tension de la bateria esta dentro de los limites

Tension de bateria medida = 53096 < 57600 Tension maxima carga?

Verificacion completa y sin errores

Error final de verificacion = 0

MODO CARGA

Se habilita la medicion para controlar el desequilibrio

La celda medida con menos tension es la 8

La celda medida con menos tension es la 9

La celda medida con menos tension es la 10

La celda medida con menos tension es la 11

La celda medida con menos tension es la 12

La celda medida con menos tension es la 13

La celda medida con menos tension es la 14

La celda medida con menos tension es la 15

Se habilita el calibrado del bloque B

Se habilita el MOSFET de la celda 8

Se habilita el MOSFET de la celda 9

Se habilita el MOSFET de la celda 10

Se habilita el MOSFET de la celda 11

Se habilita el MOSFET de la celda 12

Se habilita el MOSFET de la celda 13

Se habilita el MOSFET de la celda 14

Se habilita el MOSFET de la celda 15

Se enciende la barra de LED's con movimiento continuo indicando carga
Entramos a MOSFET_enable!
La variable DCHG es 0
La variable CHG es 1
Por tanto, solo se permite la carga
Equipo maestro encendido
Sensores de temperatura inicializados
Corriente = -10416
Tension total de la bateria = 53096
Datos del ADC leidos
Se procede a la verificacion de los limites de los sensores
Sensores de temperatura:
Primer sensor (ambiente) conectado y midiendo 25.000000
Por tanto, ventilador igual a 0
Error T2 = 0
Error T3 = 0
Error T4 = 0
No hay fallo de temperatura, se continua verificando. ErrorT = 0
Voltaje celda 0 = 3336
Voltaje celda 8 = 3301
Voltaje celda 1 = 3336
Voltaje celda 9 = 3301
Voltaje celda 2 = 3336
Voltaje celda 10 = 3301
Voltaje celda 3 = 3336
Voltaje celda 11 = 3301
Voltaje celda 4 = 3336
Voltaje celda 12 = 3301
Voltaje celda 5 = 3336
Voltaje celda 13 = 3301
Voltaje celda 6 = 3336
Voltaje celda 14 = 3301
Voltaje celda 7 = 3336

Voltaje celda 15 = 3301

La corriente instantanea = -10416mA

La corriente esta dentro de los limites

Se comprueba que la tension de la bateria esta dentro de los limites

Tension de bateria medida = 53096 < 57600 Tension maxima carga?

Verificacion completa y sin errores

Error final de verificacion = 0

MODO CARGA

Se habilita la medicion para controlar el desequilibrio

La celda medida con menos tension es la 8

La celda medida con menos tension es la 9

La celda medida con menos tension es la 10

La celda medida con menos tension es la 11

La celda medida con menos tension es la 12

La celda medida con menos tension es la 13

La celda medida con menos tension es la 14

La celda medida con menos tension es la 15

Se habilita el calibrado del bloque B

Se habilita el MOSFET de la celda 8

Se habilita el MOSFET de la celda 9

Se habilita el MOSFET de la celda 10

Se habilita el MOSFET de la celda 11

Se habilita el MOSFET de la celda 12

Se habilita el MOSFET de la celda 13

Se habilita el MOSFET de la celda 14

Se habilita el MOSFET de la celda 15

Se enciende la barra de LED's con movimiento continuo indicando carga

Entramos a MOSFET_enable!

La variable DCHG es 0

La variable CHG es 1

Por tanto, solo se permite la carga

}

FINALIZACIÓN DE LA SIMULACIÓN 3