

---

# Monitoreo y gestión de bancos de batería Liiion utilizando el integrado BQ76942 aplicado a vehículos eléctricos ligeros (LEVs)

---

Marco Tulio Trujillo Lara



UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



**Monitoreo y gestión de bancos de batería Li-ion utilizando el  
integrado BQ76942 aplicado a vehículos eléctricos ligeros  
(LEVs)**

Trabajo de graduación presentado por Marco Tulio Trujillo Lara para  
optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2024



UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



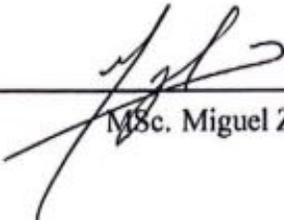
**Monitoreo y gestión de bancos de batería Li-ion utilizando el  
integrado BQ76942 aplicado a vehículos eléctricos ligeros  
(LEVs)**

Trabajo de graduación presentado por Marco Tulio Trujillo Lara para  
optar al grado académico de Licenciado en Ingeniería Mecatrónica

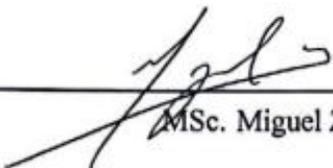
Guatemala,

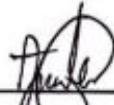
2024

Vo.Bo.:

(f)   
MSc. Miguel Zea

Tribunal Examinador:

(f)   
MSc. Miguel Zea

(f)   
Ing. Jonathan Mansilla

(f)   
Ing. Francisco Gómez

Fecha de aprobación: Guatemala, 6 de enero de 2024.

Quiero dedicar este trabajo a todas las personas que me han acompañado a lo largo de este viaje para lograr cumplir mi meta de obtener un título académico como Ingeniero. Especialmente quiero agradecer a Dios por bendecirme con la oportunidad de cursar mis estudios y luego a mi familia quienes nunca dudaron de mis capacidades y que a pesar de las dificultades nunca han dejado de alentarme en lograr esta meta.

Agradezco a mi primo Chonito ya que vivir junto a él estos años ha sido muy divertido y gracias a su compañía siempre me sentí como en casa, a pesar de haber tenido que abandonar el hogar donde crecí desde la niñez en mi pueblo Quezaltepeque, Chiquimula.

Agradezco también a mis dos mejores amigos porque ellos han estado junto a mi tanto en mis mejores momentos como en los más dificultosos, gracias por brindarme su amistad incondicional Cristopher y Alejandro, sigamos generando muchos más recuerdos juntos.

Finalmente gracias a todas aquellas personas con quienes en algún momento compartimos algún recuerdo o me tendieron la mano cuando lo llegue a necesitar.

<b>Prefacio</b>	<b>III</b>
<b>Lista de figuras</b>	<b>VII</b>
<b>Resumen</b>	<b>VIII</b>
<b>Abstract</b>	<b>IX</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Antecedentes</b>	<b>2</b>
2.1. Rover UVG . . . . .	2
2.2. Diseño de un BMS utilizando Arduino nano para baterías Li-ion . . . . .	3
2.3. Consideraciones de diseño para un BMS (Li-on) para el satélite STSAT-3 . . . . .	4
2.4. Monitoreo del estado de carga, vida y funcionamiento de baterías para vehículos eléctricos (EV) . . . . .	5
<b>3. Justificación</b>	<b>6</b>
<b>4. Objetivos</b>	<b>7</b>
4.1. Objetivo general . . . . .	7
4.2. Objetivos específicos . . . . .	7
<b>5. Alcance</b>	<b>8</b>
<b>6. Marco teórico</b>	<b>9</b>
6.1. Kit de evaluación BQ76942EVM . . . . .	9
6.2. Battery Management System (BMS) . . . . .	10
6.2.1. ¿Qué debería lograr un BMS? . . . . .	10
6.2.2. Terminología básica sobre baterías . . . . .	10
6.2.3. Estructura de una batería electroquímica convencional . . . . .	11
6.3. Proceso de carga en baterías Li-Ion . . . . .	11
6.4. Battery Management Studio (BQSTUDIO) . . . . .	12

<b>7. Desarrollo de Firmware: Escritura y lectura a registros del integrado BQ76942</b>	<b>13</b>
7.1. Microcontrolador seleccionado . . . . .	13
7.2. Instalación de bqStudio y validación el correcto funcionamiento de BQ76942EVM	14
7.2.1. Instalación . . . . .	14
7.2.2. Validación de funcionamiento del modulo de evaluación . . . . .	15
7.3. Desarrollo de la librería para el manejo del integrado BQ76942 . . . . .	17
7.3.1. Lectura de datos entre ESP32 y BQ76942 . . . . .	17
7.3.2. Lectura de registros especiales . . . . .	19
7.3.3. Escritura a registros de configuración del integrado BQ76942 . . . . .	20
<b>8. Metodología para el diseño de un banco de baterías Li-Ion</b>	<b>23</b>
8.1. Calcular los requerimientos energéticos del dispositivo . . . . .	24
8.2. Seleccionar las celdas que se utilizarán en el banco de baterías. . . . .	25
8.3. Calcular la cantidad de celdas requeridas . . . . .	25
8.4. ¿Qué hacer en caso de que no sea posible seleccionar un tipo de celda y se deba utilizar un modelo en específico? . . . . .	28
8.5. Caracterización de celdas Li-Ion . . . . .	28
8.5.1. Cargar celdas . . . . .	29
8.5.2. Encontrar el coeficiente C, capacidad y voltaje de corte de la celda . .	30
8.5.3. Calcular la cantidad de celdas requeridas con base en las características encontradas durante el proceso de caracterización . . . . .	33
8.5.4. Comparativa de costos y tamaños entre los dos bancos de baterías propuestos . . . . .	35
8.6. Configuración del integrado . . . . .	36
<b>9. Conclusiones</b>	<b>38</b>
<b>10.Recomendaciones</b>	<b>39</b>
<b>11.Bibliografía</b>	<b>40</b>
<b>12.Anexos</b>	<b>42</b>
12.1. Repositorio a la librería para el manejo del integrado BQ76942 . . . . .	42
12.2. Tablas con direcciones de registros . . . . .	42

---

## Lista de figuras

---

1.	Requerimientos energéticos para cada uno de los componentes electrónicos del Rover [1]. . . . .	2
2.	Módulos de potencia y control [1]. . . . .	3
3.	Ensamblaje de prototipo BMS con Arduino Nano [2]. . . . .	3
4.	Diagrama de la estructura general del sistema [3]. . . . .	4
5.	Diagrama de bloques del sistema de alimentación y control[3]. . . . .	4
6.	Diagrama de bloques de la solución propuesta [4]. . . . .	5
7.	Prototipo ensamblado [4]. . . . .	5
8.	Sistema de potencia y control actual del proyecto Rover UVG [1]. . . . .	6
9.	Placa de desarrollo BQ76942EVM [5]. . . . .	9
10.	Estructura de una celda electroquímica convencional. [6] . . . . .	11
11.	Curvas de carga en celdas Li-Ion [6]. . . . .	12
12.	Interfaz del programa BQSTUDIO [7]. . . . .	12
13.	Sección de descargas del software bqStudio. . . . .	14
14.	Pestaña de visualización de registros del integrado. . . . .	16
15.	Pestaña de visualización de registros de configuración del integrado. . . . .	16
16.	Primer código utilizado para la lectura de registros del integrado BQ76942. . . . .	17
17.	Diagrama de flujo que describe el proceso requerido para realizar una lectura de registros. . . . .	18
18.	Primer código utilizado (no funcional) para la escritura de registros del integrado BQ76942. . . . .	20
19.	Valores de registros de memoria antes de realizar la modificación. . . . .	21
20.	Segundo código utilizado (funcional) para la escritura de registros del integrado BQ76942. . . . .	21
21.	Valores de registros de memoria luego de realizar la modificación. . . . .	22
22.	Demanda de potencia del Rover UVG. . . . .	24
23.	Celda seleccionada para el diseño de un banco de baterías para el Rover UVG [18]. . . . .	25
24.	Disposición de celdas para el Rover UVG. . . . .	27
25.	Dispositivo utilizado para cargar las celdas 18650 [19]. . . . .	29

26.	Carga electrónica para realizar pruebas de descarga [20]. . . . .	30
27.	Montaje de todos los elementos necesarios para realizar las pruebas de descarga. Nota: Es necesario cerrar la bolsa térmica antes de iniciar con las pruebas.	31
28.	Tablas con información obtenida durante 3 distintas pruebas a diferentes tasas de descarga. . . . .	32
29.	Ejemplo de las conexiones en un banco de baterías con una distribución 4S4P.	34
30.	Comparativa de bancos de batería. . . . .	35
31.	Código utilizado para cambiar la configuración del integrado a las requeridas para el banco de baterías diseñado para el Rover UVG. . . . .	37

Este trabajo tiene como objetivo principal conocer las capacidades y funcionalidades del integrado BQ76942, para lo cual se desarrolló una librería para el microcontrolador ESP32 en un framework de Arduino para desempeñar las funciones de lectura y escritura a los registros del integrado. Se planteó una metodología de diseño para bancos de baterías Li-Ion y se validó dicha metodología diseñando un banco adecuado para el proyecto Rover UVG, el cual requiere de al menos 50 Wh para lograr una autonomía de 20 min. El banco propuesto utiliza celdas B30 en una configuración 5S1P y logra almacenar hasta 55 Wh.

---

## Abstract

---

The main objective of this work is to know the capabilities and functionalities of the BQ76942 integrated circuit, to achieve this a library was developed for the ESP32 microcontroller in the Arduino framework to perform the functions of reading and writing to the registers of the integrated circuit, design methodology for Li-Ion battery banks was proposed and this methodology was validated by designing a suitable bank for the Rover UVG project, which requires at least 50 Wh to achieve an autonomy of 20 min. The proposed bank uses B30 cells in a 5S1P configuration and can store up to 55 Wh.

# CAPÍTULO 1

---

## Introducción

---

En la actualidad la utilización de celdas de iones de litio se ha convertido en una de las tecnologías predilectas a utilizar en dispositivos recargables, sin embargo, estas pueden llegar a representar un riesgo elevado para los equipos, incluso podrían llegar a representar riesgos para los usuarios ya que este tipo de baterías son capaces de generar incendios o explosiones en los casos más extremos, además en ocasiones liberan gases tóxicos como el fluoruro de hidrógeno y el monóxido de carbono. Es por ello que se considera casi obligatorio contar con un monitor de baterías en equipos que utilicen esta tecnología, como por ejemplo, en los vehículos eléctricos ligeros (LEVs).

Este trabajo tiene como objetivo principal conocer las capacidades y funcionalidades del integrado BQ76942 el cual es un monitor de bancos de baterías de alta precisión. Para ello se utilizó un microcontrolador ESP32 para leer los registros que contienen información importante sobre el estado del banco de baterías, así mismo se utilizó este microcontrolador para escribir a los registros de configuración del integrado.

En el capítulo 7 se presenta el desarrollo de una librería para el ESP32 que contiene distintas funciones a nivel de usuario (fácil implementación) para poder realizar las lecturas y escrituras de los registros de mayor interés del integrado BQ76942 mediante el protocolo de comunicación I2C.

Así mismo, en el capítulo 8 se planteó una metodología para el diseño de bancos de baterías Li-ion utilizando herramientas accesibles en el mercado nacional para la caracterización del banco y haciendo uso del integrado BQ76942 para desempeñar el trabajo de monitoreo del banco de baterías, además se validó la metodología propuesta diseñando un banco de baterías para el Rover UVG ya que anteriormente este vehículo eléctrico ligero (LEV) funcionaba con base en baterías de ácido plomo convencionales y convertidores de voltaje de tipo buck.

Como se menciona anteriormente el uso de tecnología Li-Ion se encuentra muy extendido en la actualidad, a continuación se muestran distintos proyectos en los cuales se utilizaron algún tipo de estas baterías.

## 2.1. Rover UVG

En [1] se buscaba diseñar el sistema mecánico para el movimiento del Rover UVG, así como el módulo de alimentación que brindara la potencia necesaria para el funcionamiento de motores, sensores y otros componentes electrónicos. Se puede observar en la Figura 1 los requerimientos de voltaje y corriente estimados:

Descripción	Voltaje	Corriente nominal (estimada)
Motores	12V	10 amperios
Controlador Central	5V	1 amperio
Controlador ROS	5V	3.5 amperios
Modulo DWM	5V	0.3 amperios
Cámaras	5V	1 amperio

Figura 1: Requerimientos energéticos para cada uno de los componentes electrónicos del Rover [1].

Así mismo, en la Figura 2 se presenta un diagrama con la distribución de los módulos que requieren alimentación y los elementos que lo componen. Para suplir estas necesidades se utilizó un sistema de dos baterías de ácido-plomo para dispositivos UPS comerciales de 12V conectadas en paralelo y para la alimentación de los sistemas de control se utilizaron convertidores tipo “*buck stepdown*” los cuales utilizaron un integrado XL4015e para realizar una conversión de 12V a 5V.



Figura 2: Módulos de potencia y control [1].

## 2.2. Diseño de un BMS utilizando Arduino nano para baterías Li-ion

Los autores de [2] buscaban diseñar un sistema BMS con capacidades de monitorear, balancear y proteger un banco de baterías elaborado a partir de 3 celdas Panasonic 18650BE en configuración 3S1P. Esto con el objetivo de garantizar un manejo seguro de los paquetes de baterías para evitar daños y prolongar su vida útil. Para llevar este proyecto a cabo se utilizó un Arduino Nano como el encargado del control, y los integrados ADS1115 para sensar valores analógicos de voltaje y transformarlos en valores digitales. La información es desplegada en una pantalla para conocer los valores de voltaje de la batería, el balanceo de cargas inicia cuando una de las celdas alcanza el voltaje requerido y el exceso de cargas se drena por medio de una resistencia.

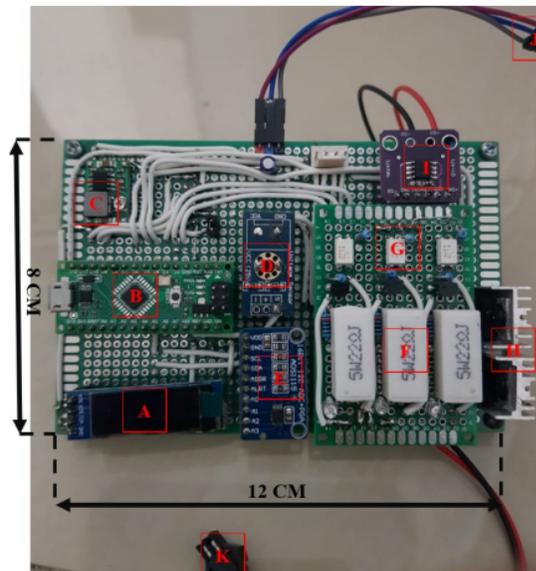


Figura 3: Ensamblaje de prototipo BMS con Arduino Nano [2].

## 2.3. Consideraciones de diseño para un BMS (Li-on) para el satélite STSAT-3

En [3] se buscó diseñar un BMS para el satélite STSAT-3 y poder distribuir energía necesaria para los sistemas mecánicos y de control de dicho dispositivo. El paquete de baterías fue elaborado a partir de 8 celdas, cada una con una capacidad de 30 Ah, voltajes nominales que oscilaban entre 3.0V-4.1V y la configuración seleccionada fue de 8S1P; es decir, el voltaje de la batería oscilaba entre 24V-32.8V. Para recargar dichas baterías se utilizaba un sistema de paneles solares.

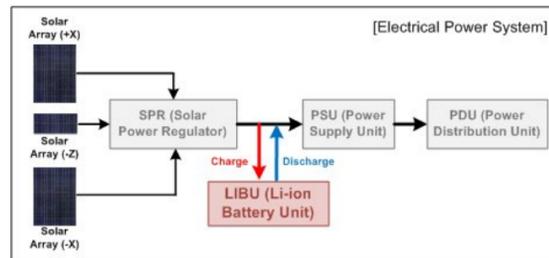


Figura 4: Diagrama de la estructura general del sistema [3].

El módulo de control de baterías fue el encargado de obtener la información necesaria para calcular el estado de carga y de salud de la batería. Este módulo estaba encargado de medir:

- Voltaje y corriente de la batería durante los periodos de carga y descarga.
- Temperatura de cada una de las celdas.
- Determinar el estado de cada celda y si hay alguna probabilidad de que esta falle.
- Proteger otros módulos de potenciales fallos como corto circuitos, sobre corriente, sobre voltaje y sobre temperatura.

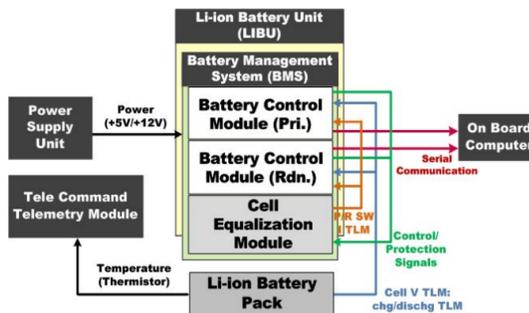


Figura 5: Diagrama de bloques del sistema de alimentación y control[3].

Como un detalle importante destaca la utilización de dos módulos de control para baterías en lugar de uno solo para tener una mayor seguridad de funcionamiento ya que se posee

un módulo de respaldo en caso de que el módulo principal presente una falla, así mismo para un sistema de doble validación de información en caso de que exista alguna duda sobre la precisión de la información recibida proveniente del módulo principal.

## 2.4. Monitoreo del estado de carga, vida y funcionamiento de baterías para vehículos eléctricos (EV)

El objetivo de [4] fue monitorear tres factores clave: estado de carga (SOC), estado de vida (SOH), y estado de funcionamiento (SOF) de un banco de baterías de tecnología Li-Ion. Para esto, se utilizó un sensor de corriente que mide la corriente suministrada por la batería y un sensor de voltaje que mide el voltaje en tiempo real de la batería. El encargado de procesar esta información fue el módulo “battery state monitor” el cual luego se comunicaba con una placa de desarrollo de Arduino, la que a su vez se encargaba de mostrar toda la información en una pantalla.

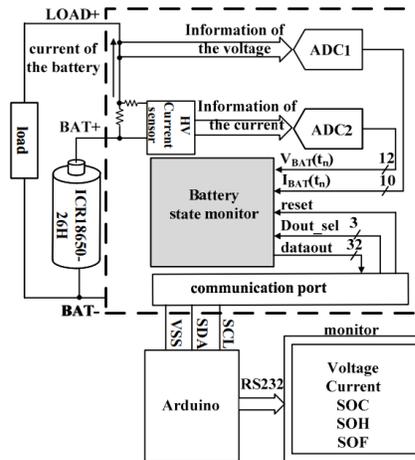


Figura 6: Diagrama de bloques de la solución propuesta [4].

La tecnología utilizada en este proyecto para diseñar el monitor de baterías fue la de un FPGA y el modelo propuesto demostró obtener un porcentaje de error en la lectura del SOC menor al 1% lo cual convierte este diseño en un prototipo de interés.

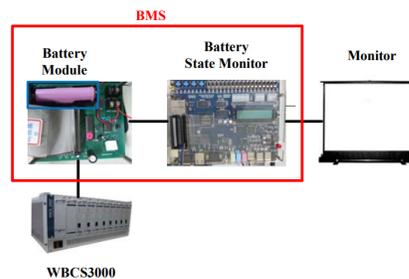


Figura 7: Prototipo ensamblado [4].

A pesar que en [1] se propuso la utilización de baterías de iones de litio para suministrar la energía necesaria para los sistemas de control y potencia del Rover UVG, no se llevó a cabo su implementación. Puede observarse en la Figura 8 que, hasta el momento, en la infraestructura del Rover no se ha implementado ningún tipo de BMS ya que en lugar de baterías de iones de litio se utilizaron baterías de ácido plomo (SLA).

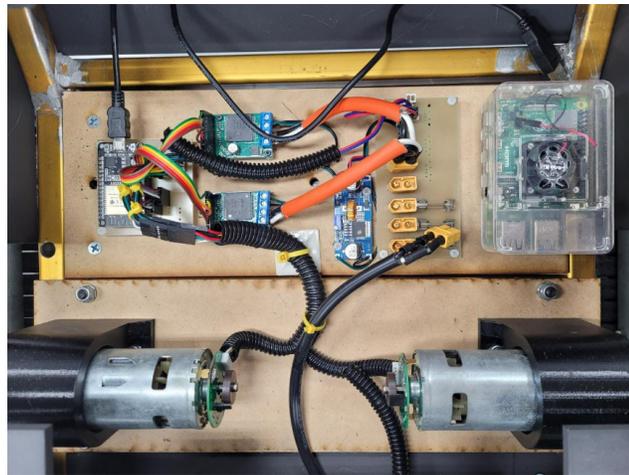


Figura 8: Sistema de potencia y control actual del proyecto Rover UVG [1].

Las baterías SLA presentaron dos limitaciones principales: la primera fue que la capacidad de descarga de las baterías no era suficiente para suministrar la corriente adecuada a los motores encargados de operar las llantas del vehículo, limitando su capacidad de movimiento, la segunda fue que el nivel de autonomía era muy bajo, aproximadamente 15 min.

Es por eso que en este trabajo de graduación se busca hacer posible una mejora en el sistema energético del proyecto Rover UVG, estableciendo una metodología que permita desarrollar bancos de baterías Li-Ion seguros de implementar y con la capacidad de entregar la potencia suficiente para lograr un mejor desempeño en la movilidad de este LEV.

### 4.1. Objetivo general

Utilizar el integrado BQ76942 de Texas Instruments para monitorear y gestionar un banco de baterías Li-ion que pueda utilizarse en un vehículo eléctrico ligero (LEV).

### 4.2. Objetivos específicos

- Desarrollar firmware que permita la comunicación entre el integrado BQ76942 y un microcontrolador.
- Obtener mediciones de parámetros que permitan evaluar y diagnosticar el correcto funcionamiento de un banco de baterías Li-ion.
- Validar la metodología propuesta mediante el diseño de un banco de baterías adecuado para el proyecto Rover UVG.

El alcance de este trabajo abarcó el desarrollar un *firmware* que permita la comunicación entre el integrado BQ76942 y el microcontrolador ESP32, utilizando el *framework* de Arduino para hacer uso de librerías ya existentes tales como la librería *Wire*.

Parte del desarrollo del firmware consiste en incluir una librería con funciones que permitan la lectura de los registros que contengan información importante para conocer el estado del paquete de baterías el cual fue simulado mediante el módulo de evaluación [5].

También se consideró incluir funciones a la librería que fueran capaces de modificar los registros de configuración del integrado.

Además, plantear una metodología para llevar a cabo el diseño de bancos de baterías Li-ion utilizando herramientas accesibles en el mercado nacional para la caracterización de las celdas y haciendo uso del integrado BQ76942 para desempeñar el trabajo de monitoreo.

Por último, validar la metodología diseñando un banco de baterías que se considere adecuado para el proyecto Rover UVG el cual en teoría podría cumplir con los requerimientos de potencia para los distintos módulos del vehículo eléctrico ligero (LEV).

### 6.1. Kit de evaluación BQ76942EVM

El módulo BQ76942EVM, el cual se puede observar en la Figura 9, es un kit de evaluación completo para poner a prueba las funcionalidades del integrado BQ76942 el cual es capaz de soportar una configuración mínima de 3 celdas en serie hasta un máximo de 10 celdas de baterías Li-Ion. Cuenta con los elementos necesarios para proteger a la batería de sobrecargas, sobre-descargas, corto circuitos, excesos de corrientes de descarga, calentamientos excesivos o temperaturas muy bajas [5].

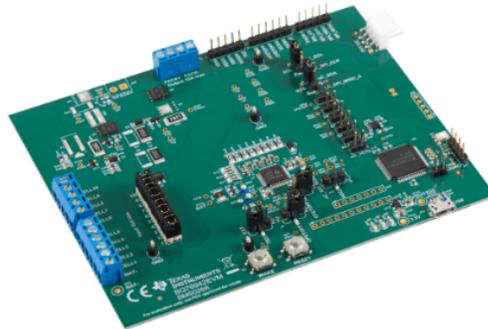


Figura 9: Placa de desarrollo BQ76942EVM [5].

## 6.2. Battery Management System (BMS)

Un sistema de manejo de baterías o BMS es un sistema embebido que se encarga del monitoreo y gestión de un paquete de baterías. Se encarga de controlar la carga, descarga y corrientes de operación del paquete de baterías, así mismo, se encarga de monitorear parámetros como el estado de carga de la batería, la temperatura a la que se encuentra y si todas las celdas se encuentran operando de forma óptima [6].

### 6.2.1. ¿Qué debería lograr un BMS?

- Proteger la seguridad humana y de los dispositivos al detectar mal-funcionamiento de un paquete de baterías y tomar acciones preventivas.
- Prolongar la vida útil de las baterías.
- Mantener a los paquetes de baterías en un estado en el cual logren cumplir los propósitos de su diseño [6].

### 6.2.2. Terminología básica sobre baterías

- **Celda:** Unidad electroquímica más pequeña que entrega un voltaje dependiendo de su composición. Pueden ser *primarias*, las cuales pueden utilizarse solamente una vez o bien *secundarias*, también conocidas como recargables.
- **Batería:** Una batería, o también llamado paquete de baterías, está compuesta por un grupo de celdas conectadas en serie, paralelo o, en muchas ocasiones, una mezcla de ambas.
- **Voltaje nominal (V):** Por lo general está impreso en la celda o batería ya que es un valor de fábrica, su magnitud puede variar cuando la batería se encuentra bajo demanda.
- **Capacidad nominal (Ah):** Especifica la cantidad de carga que la batería es capaz de almacenar.
- **Coficiente C:** Es una medición relativa de la corriente que es capaz de entregar una batería. Especifica la cantidad de corriente (A) que la batería puede suministrar de forma continua con relación a su capacidad nominal (Ah). Por ejemplo, una batería cuya capacidad de carga es de 20Ah que tenga una denominación de 1C es capaz de entregar 20A de corriente de forma continua durante 1h. Por otro lado, si se utilizara una batería de 20Ah cuya denominación sea de 10C esta es capaz de suministrar una corriente de 200A durante 10 min antes de descargarse.
- **Energía y potencia:** La capacidad total (Wh) de energía que una batería puede almacenar se calcula fácilmente multiplicando su voltaje y capacidad nominal. La potencia (W) es la capacidad que tiene dicha batería de entregar energía.

- **Celdas conectadas en serie:** En esta configuración el voltaje de la batería será la suma de los voltajes nominales de cada celda. La capacidad será la misma que la de una sola celda.
- **Celdas conectadas en paralelo:** En esta configuración capacidad de la batería será la suma de las capacidades nominales de cada celda. El voltaje será el mismo que el voltaje nominal de una sola celda [6].

### 6.2.3. Estructura de una batería electroquímica convencional

Tanto las celdas electroquímicas primarias como las celdas Li-Ion (secundarias) comparten la siguiente estructura general:

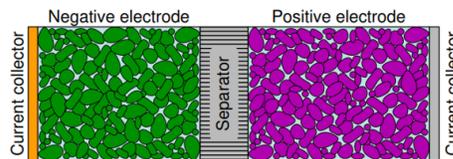


Figura 10: Estructura de una celda electroquímica convencional. [6]

La cual consta de un electrodo negativo, un electrodo positivo, un separador que evita un corto circuito entre los electrodos, un material electrolítico para el transporte de electrones y colectores de corriente que son los medios externos donde se conectan las cargas.

### 6.3. Proceso de carga en baterías Li-Ion

Los fabricantes de las baterías publican el rango de voltajes de operación de sus productos, por lo que sobrecargar o descargar demasiado una celda puede causar daño irreversible y riesgos de explosión o ignición. Por lo regular las celdas son cargadas mediante uno de dos métodos:

- **Constant current/Constant voltage (CC/CV):** Las celdas se cargan bajo una corriente constante hasta que alcanzan el voltaje de operación máximo permitido, cuando esto sucede la corriente se regula para que el voltaje de carga se mantenga estable hasta cargar por completo la celda.
- **Constant power/Constant voltage (CP/CV):** Las celdas se cargan bajo una potencia constante hasta que alcanzan el voltaje máximo permitido, cuando esto sucede nuevamente el sistema se regula para mantener constante dicho voltaje cambiando la potencia [6].

(Para observar un ejemplo gráfico puede consultarse la Figura 11)

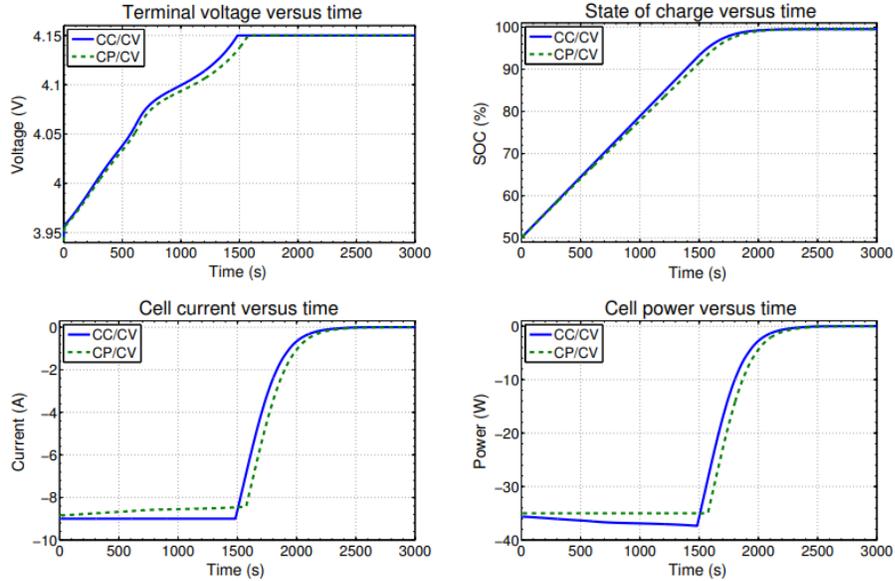


Figura 11: Curvas de carga en celdas Li-Ion [6].

## 6.4. Battery Management Studio (BQSTUDIO)

Programa de desarrollo de la compañía Texas Instruments que contiene múltiples herramientas para asistir en el proceso de evaluación, diseño, configuración, prueba y utilización de los productos que dicha compañía ofrece para sistemas de manejo de baterías. Se puede observar en la Figura 12 la pantalla de inicio del software.



Figura 12: Interfaz del programa BQSTUDIO [7].

Entre estas herramientas se incluyen acceso directo a los registros y memoria de los integrados, capacidad de monitorear a tiempo real, una interfaz amigable que facilita el envío de comandos así como guías de configuración y calibración [7].

---

## Desarrollo de Firmware: Escritura y lectura a registros del integrado BQ76942

---

Capítulo relacionado a los objetivos específicos:

- Desarrollar firmware que permita la comunicación entre el integrado BQ76942 y un microcontrolador.
- Obtener mediciones de parámetros que permitan evaluar y diagnosticar el correcto funcionamiento de un banco de baterías Li-Ion.

En este capítulo se detalla el procedimiento que se llevó a cabo para lograr realizar lecturas a los diferentes registros del integrado BQ76942 así como la escritura a registros de configuración por medio de un microcontrolador ESP32 como dispositivo maestro y cuáles métodos no fueron efectivos para evitar que el lector intente utilizarlos. Así mismo, se exponen las características actuales de la librería y sus principales funciones.

### 7.1. Microcontrolador seleccionado

El controlador que fue seleccionado para la realización de la librería para el integrado fue el ESP32 debido a su versatilidad y alta disponibilidad en el mercado nacional, además de que en la actualidad posee múltiples librerías oficiales que facilitan el proceso de programación.

Otra de las características tomadas en consideración al momento de haber seleccionado este microcontrolador, son las capacidades de *Internet of Things (IoT)* que posee ya que cuenta con capacidades WiFi y Bluetooth.

## 7.2. Instalación de bqStudio y validación el correcto funcionamiento de BQ76942EVM

El software bqStudio fue la herramienta utilizada para validar que hubo un correcto funcionamiento de la comunicación entre el integrado BQ76942 y un microcontrolador al poder visualizar directamente los registros del integrado. En otras palabras, este programa fue útil para saber si el microcontrolador era capaz de leer los registros del integrado con información importante sobre el banco de baterías. Así mismo, se utilizó para asegurarse de que el microcontrolador fuera capaz de sobre-escribir los distintos registros de configuración que posee el integrado.

Un segundo punto de interés era el de validar que el módulo de evaluación BQ76942EVM funcionara correctamente antes de realizar pruebas en la comunicación entre el integrado y el microcontrolador.

### 7.2.1. Instalación

Para realizar la instalación del programa se consultó la página oficial [7] del fabricante y luego se descargó el instalador de la última versión de prueba (opción resaltada en amarillo en la Figura 13), la razón por la cual se utilizó esta versión fue debido a que el modelo del integrado que se utilizó aún no se encuentra disponible en la versión estable del programa. Por último, se realizaron todos los pasos indicados en el instalador los cuales se resumen en aceptar el acuerdo de usuario final y especificar la ruta en la cual se instaló el programa.

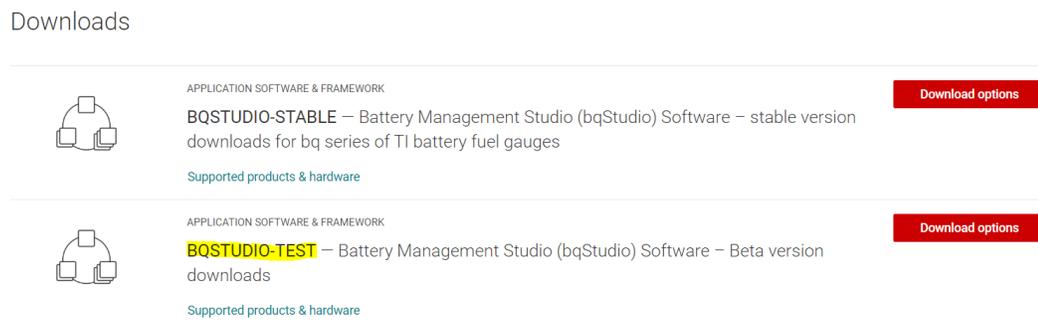


Figura 13: Sección de descargas del software bqStudio.

### 7.2.2. Validación de funcionamiento del modulo de evaluación

Al finalizar la instalación se consultó el manual de usuario del BQ76942EVM [8] el cual se obtuvo de la página oficial del producto [5] y se realizaron todos los pasos listados en la sección 2.2 *Quick start* para validar el correcto funcionamiento del módulo de evaluación. Estos se resumen a continuación:

1. Instalar puentes eléctricos (*jumpers*) de simulación de celdas de baterías. Así mismo posicionar los puentes de comunicación en la configuración: uC\_SCL y uC\_SDA.
2. Conectar mediante el cable USB el integrado BQ76942EVM y una computadora con bqStudio instalado.
3. Conectar una fuente de voltaje que sea capaz de suministrar entre 22V y 47V de tensión y una corriente de al menos 250mA. Configurar la salida aproximadamente a 36V.
4. Iniciar el programa bqStudio, el programa detectará automáticamente la placa que está conectada y se mostrarán los registros del integrado en la interfaz de usuario. Presionar el botón *Scan* para actualizar los valores continuamente.
5. En la interfaz de usuario presionar el botón de *Memory*, luego presionar el botón *Settings* y habilitar el bit *CUV* del registro de configuración llamado *Enable protections A*. En la Figura 15 se puede observar una captura de pantalla de esta pestaña.
6. Presionar el botón *FET\_EN* del panel de comandos. Recuadro rojo en la Figura 15.
7. Regresar al panel de visualización de registros y observar que los bits *CHG\_FET* y *DSG\_FET* estén encendidos (en rojo). En la Figura 14 se puede observar una captura de pantalla de esta pestaña.
8. Ajustar el voltaje de la fuente de alimentación a un valor aproximado de 45V, el bit *CHG\_FET* debería apagarse (en color verde).
9. Luego ajustar el voltaje de la fuente de alimentación a un valor aproximado de 24V, ahora el bit *DSG\_FET* debería apagarse (en color verde) y el voltaje del paquete debería caer hasta aproximadamente 0V.

Luego de haber realizado los pasos listados anteriormente se logró identificar ciertas características que se consideran importantes de mencionar, la primera es que al conectar la placa a la computadora se encendieron algunos LEDs los cuales aparentaban que el integrado ya había encendido, sin embargo, el programa bqStudio no detectará ningún dispositivo al menos que se alimente al circuito con la fuente externa especificada en el paso 3.

La segunda característica es que el integrado consume menos de 50mA cuando se está trabajando con las celdas de simulación, por lo tanto, se recomienda limitar la cantidad de corriente que entrega la fuente a solo 100mA para las pruebas, con el fin de minimizar el riesgo de que algún componente se dañe al momento de manipular la placa de pruebas y ocasionar algún falso contacto.

En la Figura 14 se pueden observar los valores de los registros de memoria, donde los bits en color verde representan cero y los bits en color rojo representan uno. Con esto se valida que el módulo está funcionando correctamente y que la comunicación entre el integrado y la computadora se logró exitosamente.

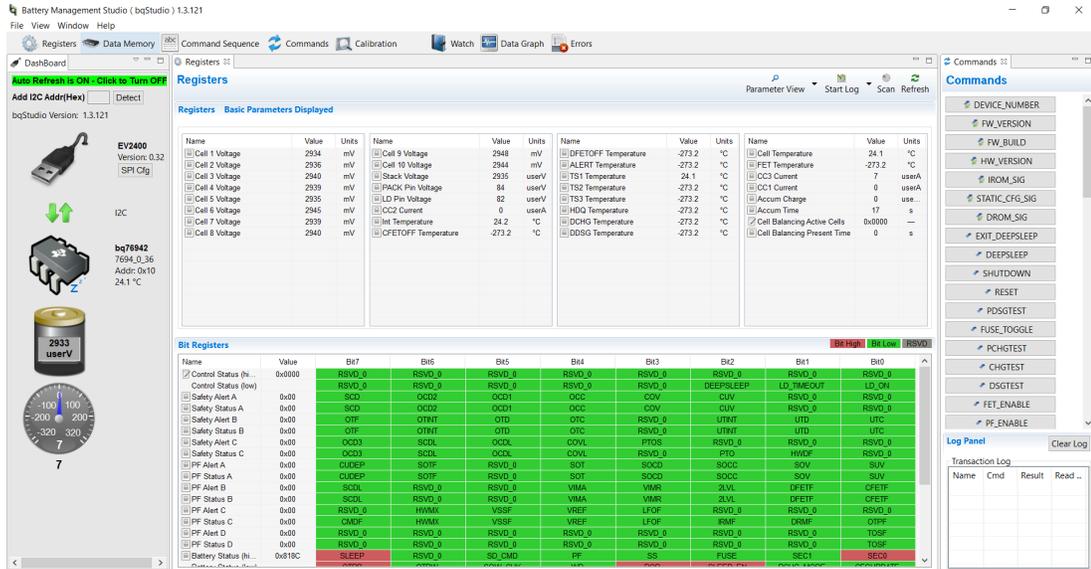


Figura 14: Pestaña de visualización de registros del integrado.

En la Figura 15 se pueden observar todos los registros de configuración, los cuales se deben modificar dependiendo de las necesidades de la aplicación en la que se fuera a utilizar el integrado.

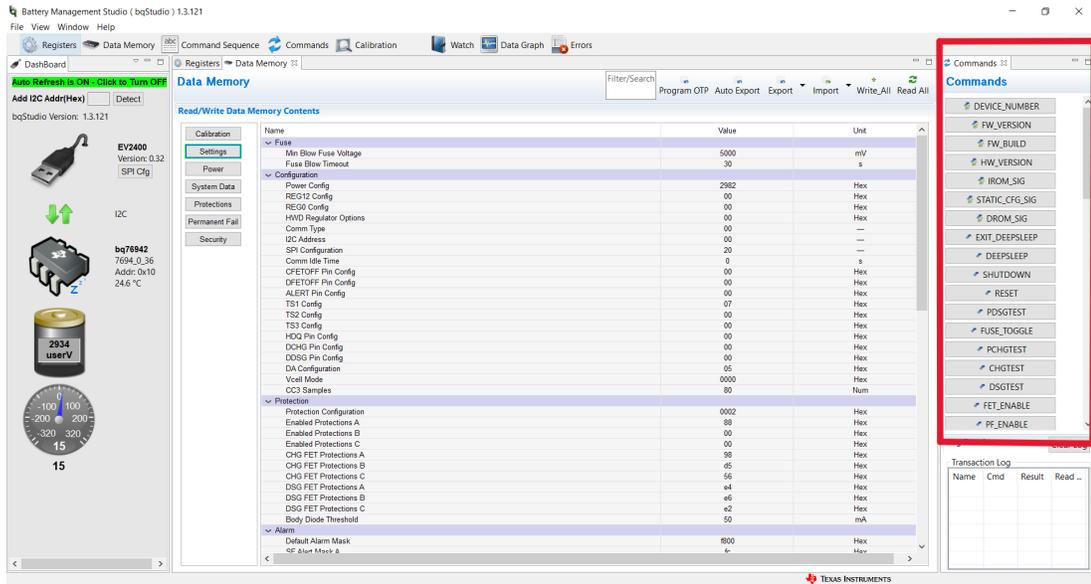


Figura 15: Pestaña de visualización de registros de configuración del integrado.

## 7.3. Desarrollo de la librería para el manejo del integrado BQ76942

### 7.3.1. Lectura de datos entre ESP32 y BQ76942

Como se mencionó anteriormente, el protocolo de comunicación que se seleccionó para la interconexión entre el integrado y el microcontrolador fue I2C ya que este protocolo da la flexibilidad de conectar múltiples integrados de ser requerido. Para llevar a cabo la implementación primero se investigó si el microcontrolador ESP32 contaba con alguna librería oficial para realizar comunicación mediante I2C y se encontró en la página oficial del proyecto *Arduino-ESP32* [9] que sí había una librería disponible para implementar esta comunicación a un nivel de usuario.

Para ello se que utilizó la librería *wire* la cual permitió utilizar funcionalidades generales del módulo I2C del microcontrolador de una forma bastante simple. Para realizar la primera prueba de lectura se realizó un código basado en el ejemplo encontrado en [10] el cual tiene la estructura de una lectura convencional de un registro mediante comunicación I2C, este código se puede observar en la Figura 16.

```
#include <Wire.h>
#include <Arduino.h>

Wire.beginTransmission( Slave );           //Inicializar la comunicación.
Wire.write( reg );                         //Ubicarse en la dirección del registro.
Wire.endTransmission();                   //Terminar comunicación.

Wire.requestFrom( Slave, N );             //Solicitar N registros al esclavo.

while ( Wire.available() ){               //Verificar si hay datos para lectura
  data = Wire.read();                     //Leer datos recibidos
  Serial.println(data, HEX);              //Imprimir datos en terminal
}
```

Figura 16: Primer código utilizado para la lectura de registros del integrado BQ76942.

Para realizar una lectura a registros de memoria del integrado los pasos que se realizaron en el microcontrolador fueron lo siguientes: Configurar la comunicación I2C, iniciar la comunicación con el esclavo, enviar la dirección del registro de interés, cerrar la comunicación con el dispositivo esclavo, solicitar el número determinado de registros que se deseen leer y por ultimo realizar la lectura de los registros solicitados. Se puede observar en la Figura 17 un diagrama de flujo con más información.

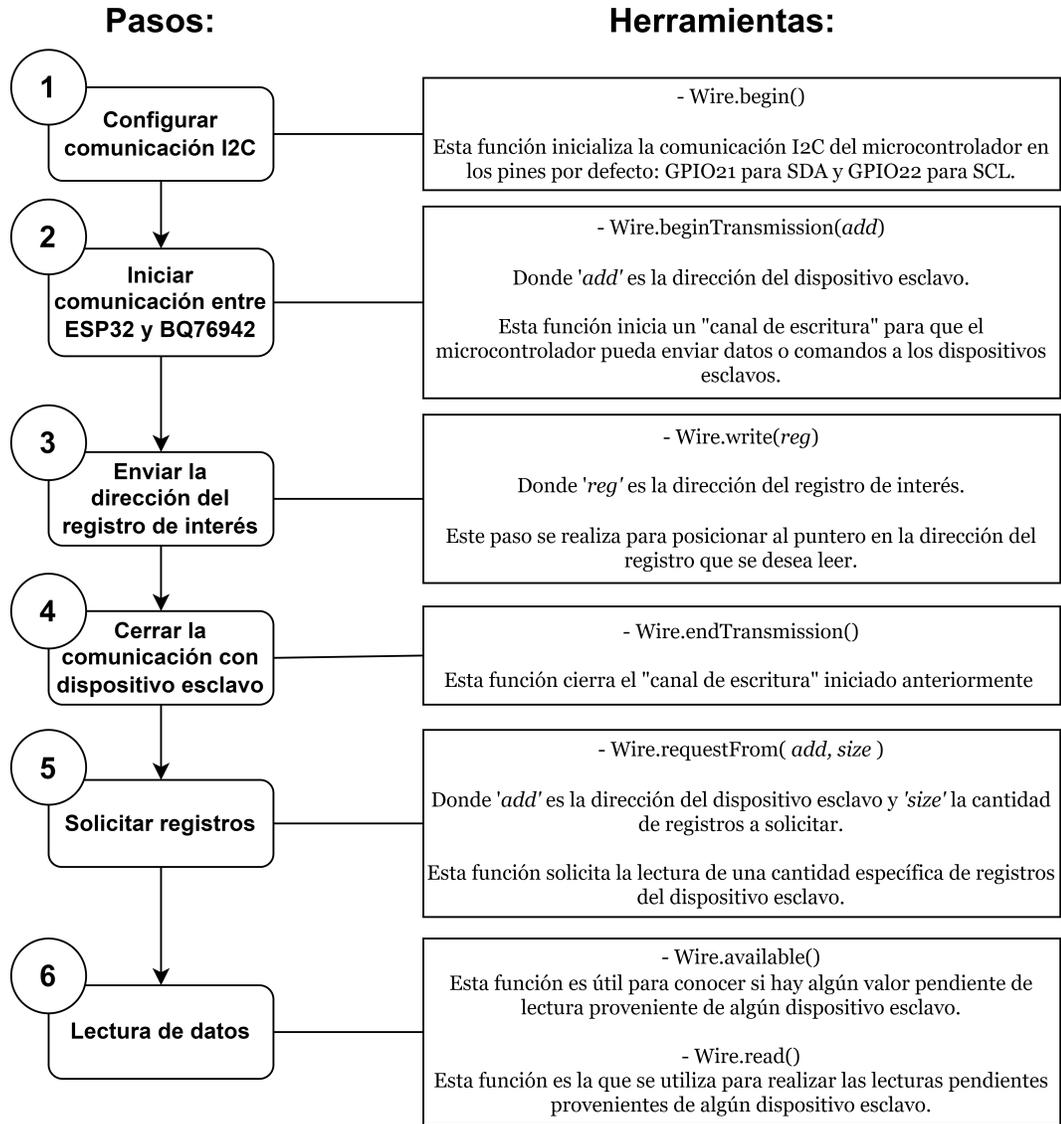


Figura 17: Diagrama de flujo que describe el proceso requerido para realizar una lectura de registros.

Al utilizar el código de la Figura 16 se lograron obtener distintos valores tales como: valores de voltajes de las distintas celdas, temperaturas de distintos elementos, valor de voltaje del paquete de baterías, entre otros. Por lo tanto, se logró realizar una operación exitosa de lectura, sin embargo, al momento de intentar realizar lectura de los registros de configuración la respuesta obtenida no fue la esperada, es decir, que no hubo una lectura exitosa.

Luego de consultar el manual técnico [11] en la sección *3.1 Direct Commands and Sub-commands* se encontró que solamente se pueden acceder de esta manera a todos los "comandos directos" listados en la sección *11.1.1. Direcciones de comandos directos* del capítulo de Anexos de este documento.

### 7.3.2. Lectura de registros especiales

Luego de consultar el manual técnico [11] se logró descubrir que el integrado tiene distintas categorías de registros, los cuales se acceden de diferente manera. Los únicos registros que pueden ser leídos de forma convencional son los ya anteriormente mencionados “comandos directos” los cuales son registros que poseen una dirección de solamente un byte, los registros “especiales” poseen una dirección de dos bytes y se dividen en las siguientes categorías:

1. **Subcommands:** Contiene información del fabricante como la identificación del integrado, versión de firmware, entre otros. Así también, almacena algunas mediciones finales de temperatura, corriente o voltaje antes de activarse cualquiera de las protecciones y algunos registros de estado. Se pueden encontrar en en la sección 11.1.3. *Tablas: Subcomandos.* del capítulo de Anexos.
2. **Command only subcommands:** Como su nombre lo indica son sub-comandos que solamente deben ser escritos al integrado para cambiar su configuración, en otras palabras, no tienen la capacidad de ser leídos ya que no almacenan información. Se pueden encontrar en en la sección 11.1.2. *Tablas: Command only Subcommands.* del capítulo de Anexos.
3. **Data memory registers:** Registros de configuración del integrado. Estos también son considerados un tipo de subcomando. Se pueden encontrar en en la sección 11.1.4. *Tablas: Registros de configuración.* del capítulo de Anexos.

Para lograr acceder a la información almacenada en los registros especiales, exceptuando a *Command only subcommands*, se debe realizar una serie de pasos descritos en la sección: 13.1 *Data Memory Access* del manual técnico [11] los cuales se resumen en lo siguiente:

1. Escribir el byte menos significativo de la dirección del subcomando al registro 0x3E.
2. Escribir el byte más significativo de la dirección del subcomando al registro 0x3F.
3. Leer nuevamente el valor en los registros 0x3E y 0x3F, si el retorno es 0xFF esto indica que aún no se ha ejecutado el subcomando. Cuando el retorno sea la dirección escrita originalmente esto indica que el subcomando se ha ejecutado.
4. Luego de que el subcomando termine de ejecutarse se deben leer los registros del 0x40 al 0x61 dependiendo del tamaño de la información del registro solicitado.

Para llevar a cabo una función capaz de leer este tipo de registros especiales se consultó la pagina oficial del producto [12] y se encontraron diferentes códigos disponibles para otros microcontroladores de Texas Instruments, los cuales se utilizaron como ejemplo para desarrollar un código para el ESP32 que fuera capaz de realizar lecturas a los registros especiales. También se encontró un recurso audiovisual en [13] proporcionado por el fabricante en el cual se explica como el funcionamiento y como utilizar las funciones proporcionadas en los códigos de ejemplo.

### 7.3.3. Escritura a registros de configuración del integrado BQ76942

La primera prueba que se realizó para tratar de escribir a los registros de configuración del integrado realizar una escritura convencional a un registro específico mediante el protocolo de comunicación I2C con ayuda del ejemplo encontrado en [14]. Sin embargo, este intento fue un fracaso ya que no hubo ningún cambio al momento de intentar sobre-escribir al registro de configuración *Power Config* cuya dirección es 0x9234. El código utilizado se puede observar en la Figura 18 y se muestra para recomendar al lector el no tratar de realizar una escritura de esta manera ya que es una forma muy extendida por la documentación disponible sobre el uso de la librería *Wire*.

A screenshot of an Arduino IDE window showing C++ code for I2C communication. The code includes `<Wire.h>` and `<Arduino.h>`. It uses `Wire.beginTransmission( Slave );` to initialize communication, `Wire.write(0x9234);` to set the register address, `Wire.write(0x2980);` to write the data, and `Wire.endTransmission();` to finish. Comments in Spanish explain each step: //Inicializar la comunicación., //Ubicarse en la dirección del registro., //Sobre escribir al registro., and //Terminar comunicación.

```
#include <Wire.h>
#include <Arduino.h>

Wire.beginTransmission( Slave );           //Inicializar la comunicación.
Wire.write(0x9234);                        //Ubicarse en la dirección del registro.
Wire.write(0x2980);                        //Sobre escribir al registro.
Wire.endTransmission();                   //Terminar comunicación.
```

Figura 18: Primer código utilizado (no funcional) para la escritura de registros del integrado BQ76942.

Luego que la escritura convencional mediante I2C no funcionara se consultó nuevamente el código proporcionado por el fabricante en [12] así como el manual técnico [11]. Con ayuda de estos recursos se lograron determinar los pasos necesarios para lograr sobre-escribir los registros de configuración del integrado, los cuales se enumeran a continuación:

1. Escribir el byte menos significativo de la dirección del registro a sobre-escribir a la dirección 0x3E.
2. Escribir el byte más significativo de la dirección del registro a sobre-escribir a la dirección 0x3F.
3. Escribir el nuevo valor para el registro de memoria en formato *little endian* en el búfer de transferencia (0x40 a 0x5F). Hay que notar que se pueden escribir hasta 32 bytes de información para los registros de memoria en un solo bloque de escritura.
4. Escribir la suma de verificación en el registro 0x60 y luego el largo de la información en 0x61.
5. La información puede ser verificada leyendo nuevamente el registro recién modificado escribiendo su dirección en 0x3E y 0x3F como el paso 1 y 2.
6. Leer el largo de la respuesta en el registro 0x61 ya que el búfer de transmisión del integrado se llenará con 36 bytes de datos (por lo que se debe leer solamente la cantidad de bytes obtenida en el paso anterior).
7. Comenzar la lectura de datos desde 0x40 hasta el largo de la información obtenida con anterioridad, por ejemplo, para un registro de 2 bytes se debe leer la dirección 0x40 y 0x41.

- Por último revisar la suma de verificación leyendo la dirección 0x60 para asegurar la integridad de los datos recibidos.

Con ayuda de los códigos de ejemplo proporcionados por el fabricante se lograron plantear las funciones necesarias para poder realizar los pasos listados anteriormente y lograr una escritura exitosa a los registros de configuración del integrado.

En la Figura 19 se pueden observar los valores por defecto de los registros de configuración *Power config* y *Enable protections A* con un valor de 0x2982 y 0x88 respectivamente, los cuales son los registros que se deseaban modificar durante la segunda prueba de escritura.

Name	Value	Unit
▼ Fuse		
Min Blow Fuse Voltage	5000	mV
Fuse Blow Timeout	30	s
▼ Configuration		
Power Config	2982	Hex
REG12 Config	00	Hex
REG0 Config	00	Hex
HWD Regulator Options	00	Hex
Comm Type	00	—
I2C Address	00	—
SPI Configuration	20	—
Comm Idle Time	0	s
CFETOFF Pin Config	00	Hex
DFETOFF Pin Config	00	Hex
ALERT Pin Config	00	Hex
TS1 Config	07	Hex
TS2 Config	00	Hex
TS3 Config	00	Hex
HDQ Pin Config	00	Hex
DCHG Pin Config	00	Hex
DDSG Pin Config	00	Hex
DA Configuration	05	Hex
Vcell Mode	0000	Hex
CC3 Samples	80	Num
▼ Protection		
Protection Configuration	0002	Hex
Enabled Protections A	88	Hex
Enabled Protections B	00	Hex
Enabled Protections C	00	Hex

Figura 19: Valores de registros de memoria antes de realizar la modificación.

Luego de identificar los registros de memoria que se deseaban modificar para realizar la segunda prueba de escritura, se ejecutó el código que se puede observar en la Figura 20 en el cual se utilizan simplemente 2 funciones distintas para lograr realizar una escritura a los registros de configuración del integrado.

```

1 CommandSubcommands(SET_CFGUPDATE);           //Ingresar al modo configuración
2
3 bqSetRegister(PowerConfig, 0x2980, 2);       //Modificar el valor de <<Power Config>>
4 bqSetRegister(EnabledProtectionsA, 0xBC, 1); //Modificar el valor de <<Enable protections A>>
5
6 CommandSubcommands(EXIT_CFGUPDATE);         //Salir del modo configuración

```

Figura 20: Segundo código utilizado (funcional) para la escritura de registros del integrado BQ76942.

A continuación se presentan más detalles sobre las funciones que fueron utilizadas en la Figura 20:

- CommandSubcomands(SET\_CFGUPDATE):** La función *CommandSubcommands* sirve para enviar cualquiera de los subcomandos listados en la sección 11.1.3. *Tablas: Subcomandos* del capítulo de Anexos. El subcomando *SET\_CFGUPDATE* le indica al integrado que se van a realizar cambios en los registros de configuración, si no se realiza este paso antes de intentar realizar los cambios la operación de escritura no funcionará.
- bqSetRegister:** Esta función es la que se utilizó para sobre-escribir los registros *Power config* y *Enable protections A* es una función que permite realizar cambios a los registros de configuración, requiere de 3 parámetros: el primero indica la dirección del registro de configuración, el segundo es el nuevo valor que se desea asignar y el tercero la cantidad de bytes que tiene dicho nuevo valor.
- CommandSubcomands(EXIT\_CFGUPDATE):** El subcomando *EXIT\_CFGUPDATE* sirve para indicar al integrado que se está saliendo del modo configuración y que se deben aplicar los cambios realizados a los registros.

Luego de ejecutar el código se puede observar en la Figura 21 que se logró sobre-escribir el valor de los registros satisfactoriamente.

Name	Value	Unit
▼ Fuse		
Min Blow Fuse Voltage	5000	mV
Fuse Blow Timeout	30	s
▼ Configuration		
Power Config	2980	Hex
REG12 Config	00	Hex
REG0 Config	00	Hex
HWD Regulator Options	00	Hex
Comm Type	00	—
I2C Address	00	—
SPI Configuration	20	—
Comm Idle Time	0	s
CFETOFF Pin Config	00	Hex
DFETOFF Pin Config	00	Hex
ALERT Pin Config	00	Hex
TS1 Config	07	Hex
TS2 Config	00	Hex
TS3 Config	00	Hex
HDQ Pin Config	00	Hex
DCHG Pin Config	00	Hex
DDSG Pin Config	00	Hex
DA Configuration	05	Hex
Vcell Mode	0000	Hex
CC3 Samples	80	Num
▼ Protection		
Protection Configuration	0002	Hex
Enabled Protections A	bc	Hex
Enabled Protections B	00	Hex
Enabled Protections C	00	Hex

Figura 21: Valores de registros de memoria luego de realizar la modificación.

---

### Metodología para el diseño de un banco de baterías Li-Ion

---

En este capítulo se estableció la metodología necesaria para el diseño de un banco de baterías Li-Ion. Se listan algunas de las características que son necesarias de conocer sobre el banco de baterías para poder realizar la configuración del integrado BQ76942 y las herramientas que se recomienda utilizar para obtener dichos parámetros. Se valida dicha metodología diseñando un banco de baterías para el proyecto Rover UVG.

Los pasos a seguir en la metodología propuesta son los siguientes:

- Calcular el requerimiento energético del dispositivo para el cual se está diseñando el banco de baterías.
- Seleccionar las celdas que se utilizarán para la construcción del banco de baterías.
- Calcular la cantidad de celdas requeridas.
- ¿Qué hacer si no es posible seleccionar las celdas y se deben utilizar celdas específicas ya disponibles?
- Caracterización de las celdas para la elaboración de un banco de baterías.
- Calcular la cantidad de celdas requeridas con base en las características encontradas durante el proceso de caracterización.

## 8.1. Calcular los requerimientos energéticos del dispositivo

Para poder dimensionar el banco de baterías es necesario conocer los requerimientos de potencia y el tiempo de autonomía deseado del dispositivo en el cual se instalará el banco. En el caso del Rover UVG se utilizaron los datos que se obtuvieron en la sección de antecedentes listados en la Figura 1. Para calcular la demanda de potencia se utilizó el concepto de la Ley de Ohm, la cual establece que la potencia es el resultado de multiplicar voltaje y corriente directa [15].

$$P = VI. \quad (1)$$

Como se puede observar en la Figura 22 el rover tiene un máximo aproximado de 150 Watts de demanda en potencia.

Descripción	Voltaje	Corriente nominal	Potencia
Motores	12V	10 amperios	120W
Controlador central	5V	1 amperio	5W
Controlador ROS	5V	3.5 amperio	17.5W
Modulo DWM	5V	0.3 amperios	1.5W
Cámaras	5V	1 amperio	5W
		<b>Total aproximado</b>	<b>150W</b>

Figura 22: Demanda de potencia del Rover UVG.

Para calcular la cantidad de energía que debe almacenar el banco de baterías fue necesario conocer cuánto tiempo es requerido que el dispositivo funcione de manera continua antes de que se agote la carga. Para el proyecto Rover UVG se lograba una autonomía de 15 min con las baterías de ácido plomo, es por ello que para el diseño del nuevo banco de baterías se intentará lograr una mejora de 5 min en su capacidad, por lo tanto, para calcular la capacidad del banco de baterías se utilizará la ecuación (2) la cual se obtuvo de [16].

$$E = Pt. \quad (2)$$

Donde:

- E = Energía que se debe almacenar.
- Potencia (P) = 150 W.
- Tiempo de operación (t) = 20 min = 0.33 h.

Por lo tanto, la energía mínima que el nuevo banco de baterías debería ser capaz de almacenar es de aproximadamente 50 Wh para lograr que el rover alcance al menos 20 min de autonomía.

## 8.2. Seleccionar las celdas que se utilizarán en el banco de baterías.

En el mejor de los casos será posible seleccionar las celdas que se utilizarán en la construcción del banco de baterías, para tomar una decisión adecuada se debe tomar en consideración: requerimientos energéticos, requerimientos de voltaje y requerimientos de potencia.

Por lo tanto, siguiendo con el ejemplo del Rover UVG se conoce que el requerimiento energético es de 50Wh y el requerimiento de potencia es de 150W. Para este trabajo se buscaron opciones en electrónicas locales y se seleccionó la batería B30 encontrada en [17]:



Figura 23: Celda seleccionada para el diseño de un banco de baterías para el Rover UVG [18].

Esta celda posee las siguientes características:

- **Modelo:** B30.
- **Marca:** VapCell.
- **Voltaje nominal:** 3.7V
- **Corriente máxima estable:** 20 A (6.6 C).
- **Capacidad energética:** 3 Ah.

El motivo por el cual se seleccionó esta celda fue por su capacidad de entregar grandes cantidades de corriente, esto es necesario para poder suministrar la potencia requerida para el correcto funcionamiento de todos los módulos del Rover UVG.

## 8.3. Calcular la cantidad de celdas requeridas

Ya que se conocen las características de las celdas que se van a utilizar para la construcción del banco se debe diseñar la disposición de las celdas, es decir, la cantidad de celdas en serie y en paralelo, para ello es necesario suplir las necesidades de: *voltaje, requerimiento energético y potencia*.

Para el Rover UVG se conoce:

- El requerimiento energético es de 50 Wh.
- El voltaje más alto requerido es de 12 V para la alimentación de los motores.
- Tiene una demanda de potencia aproximada de 150 W.

Utilizando como ejemplo la celda B30 seleccionada en la sección anterior se procede a calcular la cantidad de celdas en serie y paralelo requeridas para cumplir con los requerimientos de diseño.

Para calcular la cantidad de celdas en serie se recomienda que el voltaje final del banco supere el voltaje más alto requerido por el dispositivo, en el caso del Rover UVG este serían los 12 V requeridos por los motores; por lo tanto:

$$Serie = Voltaje\ requerido / Voltaje\ celda$$

$$Serie = 12\ V / 3.7\ V = 3.24\ celdas$$

Se recomienda redondear hacia arriba, ya que cuando las celdas se descargan o están operando su voltaje disminuye. Por lo tanto:

$$Serie = 4\ celdas$$

Esto a su vez permite conocer el voltaje final del banco:

$$V_{bank} = 4\ celdas * 3.7\ V = 14.8\ V$$

Para calcular la cantidad de celdas que se requieren en paralelo es necesario hacer uso de la capacidad requerida (50 Wh para el Rover UVG) y se procede de la siguiente manera:

$$Paralelo = \frac{Capacidad\ requerida\ en\ Wh}{(Capacidad\ de\ celda\ en\ Ah)(V_{bank})}$$

$$Paralelo = \frac{50\ Wh}{(3\ Ah)(14.8\ V)} = 1.12\ celdas$$

En estos casos también se recomienda redondear hacia arriba, por lo que se necesitaría de 2 celdas en paralelo para cumplir con el requerimiento de 50 Wh, sin embargo, hacer esto puede sobredimensionar la capacidad del banco ya que la nueva capacidad sería de:

$$E = V_{bank} * E_{cell} * N_{paralelo} = (14.8\ V)(3\ Ah) * (2) = 88.8\ Wh$$

Esta nueva capacidad es un 78 % más grande que la capacidad requerida en el diseño, lo cual podría ser deseable ya que esto extiende el tiempo de autonomía del dispositivo, sin embargo, también aumenta considerablemente el costo de fabricación, por lo tanto, una solución alternativa sería la de añadir una celda más en serie para que la capacidad final del banco se acerque más a los requerimientos.

De esta manera, en lugar de armar un banco de baterías con una distribución de cuatro celdas en serie y 2 en paralelo (4S2P abreviado), se opta por recomendar una distribución de 5S1P. Con esta configuración el voltaje máximo del banco será de 18.5 V, por lo tanto es posible suministrar todos los niveles de voltaje requeridos para los distintos módulos haciendo uso de reguladores de voltaje, así mismo se cumple con la demanda de potencia y energía, ya que la potencia y capacidad máxima en esta configuración es de:

$$P = VI = (18.5 V)(20 A) = 370 W$$

$$E = (3 Ah)(18.5 V) = 55.5 Wh$$

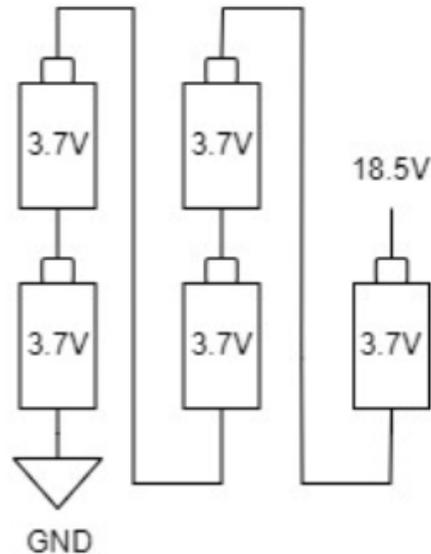


Figura 24: Disposición de celdas para el Rover UVG.

En resumen, las capacidades del banco serían las siguientes:

- Voltaje nominal (del paquete): 18.5 V.
- Corriente máxima (del paquete): 20 A.
- Potencia máxima: 370 W.
- Capacidad energética: 55.5 Wh.

## 8.4. ¿Qué hacer en caso de que no sea posible seleccionar un tipo de celda y se deba utilizar un modelo en específico?

En el caso en el que no sea posible seleccionar las celdas que serán utilizadas en el banco y se deba utilizar algún modelo en específico o celdas ya disponible existen dos posibles escenarios:

El primero, es el caso en el que las celdas están bien etiquetadas o cuenten con su propia hoja de datos, por lo tanto se conocen sus parámetros como la capacidad (mAh), el voltaje nominal (V), la corriente máxima que es capaz de suministrar (ó el coeficiente C), las temperaturas de operación, entre otros datos importantes. En este caso simplemente se realiza el mismo procedimiento que en las secciones anteriores utilizando los valores establecidos por el fabricante.

El segundo caso es un poco más complicado, ya que en este escenario las celdas que se deben utilizar no están debidamente etiquetadas, por lo tanto, se desconoce el modelo y no se sabe nada sobre sus parámetros. En este caso es necesario caracterizar manualmente las celdas, en otras palabras, se requiere de hacer pruebas para determinar valores importantes sobre sus capacidades y características.

## 8.5. Caracterización de celdas Li-Ion

Para poder utilizar los integrados BQ76942 es necesario conocer los siguientes parámetros de las celdas con las que se trabajará:

- Capacidad de la celda (Ah ó Wh).
- Voltaje máximo y mínimo de operación.
- El voltaje de terminación, que se refiere al voltaje teórico en el cual la celda ha llegado aproximadamente al 0 % de su capacidad.
- Coeficiente C de la celda o corriente máxima que puede entregar.

Para llevar a cabo una caracterización básica es necesario utilizar:

- Carga electrónica variable con capacidad para medir Ah o Wh.
- Empaquetado aislante para las baterías.
- Herramienta para medir temperatura.
- Cargador de baterías Li-Ion.

A continuación se presenta el procedimiento que se realizó para la llevar a cabo la caracterización de las celdas disponibles en el departamento de ingeniería electrónica, electromecánica y biomédica.

### 8.5.1. Cargar celdas

Como primer paso se deben cargar por completo todas las celdas, es decir, que se encuentren al 100 % de su capacidad. Para ello es necesario utilizar cualquier cargador de baterías especializado que soporte celdas Li-Ion 18650. Por ejemplo, el utilizado para las pruebas durante el desarrollo de este documento fue el cargador de baterías de la marca WISSBLUE la cual puede cargar hasta 4 celdas al mismo tiempo.



Figura 25: Dispositivo utilizado para cargar las celdas 18650 [19].

### 8.5.2. Encontrar el coeficiente C, capacidad y voltaje de corte de la celda

Este segundo paso tiene como objetivo conocer múltiples características de las celdas, las características que se busca identificar son:

- **Coeficiente C de descarga:** Indica que tanta corriente es capaz de entregar la celda de forma segura.
- **Capacidad de la celda (Ah ó Wh):** Indica la cantidad de energía que es capaz de almacenar una celda.
- **Voltaje de corte de la celda:** Voltaje específico o caída en el voltaje que indica que la celda está pronto a descargarse completamente.

Para llevar a cabo los experimentos de este segundo paso, es necesario utilizar una carga electrónica como la que se observa en la Figura 26, el módulo utilizado durante la elaboración de este trabajo fue el XY-FZ25 de 25W el cual podía drenar una corriente máxima de 4 A de una sola celda 18650.



Figura 26: Carga electrónica para realizar pruebas de descarga [20].

#### Procedimiento

Antes de comenzar con las pruebas de descarga se debe preparar la herramienta de medición de temperatura, para ello se puede utilizar un multímetro con capacidad de medir temperatura mediante una sonda térmica y adherir dicha sonda a la celda de prueba mediante el uso de cualquier tipo de cinta adhesiva, este método fue el utilizado durante la elaboración de este trabajo de graduación.

Luego se debe almacenar la celda en algún empaquetado aislante para evitar que el ambiente altere la confiabilidad de las mediciones de temperatura, puede utilizarse algo sencillo, por ejemplo, durante la elaboración de este trabajo se utilizó una bolsa térmica para alimentos.

Se deben realizar las conexiones del módulo XY-FZ25 según el fabricante, donde L+ y L- son las terminales positivas y negativas de la celda respectivamente. Luego se debe configurar la corriente de descarga, para ello se recomienda empezar por 1 C; si durante la primera prueba la celda se calienta demasiado será necesario disminuir la corriente de descarga, por el contrario, si la celda no se calienta se puede proceder a realizar otro intento con un coeficiente C mayor, se recomienda ir aumentando 0.5 C por cada iteración.

Realizar pruebas de descarga hasta encontrar la corriente (coeficiente  $C$ ) en la cual la celda alcanza temperaturas elevadas, es decir, aumentos en la temperatura que se consideren perjudiciales para las celdas 18650. De esta manera se puede determinar un coeficiente  $C$  seguro para la celda que se está caracterizando.

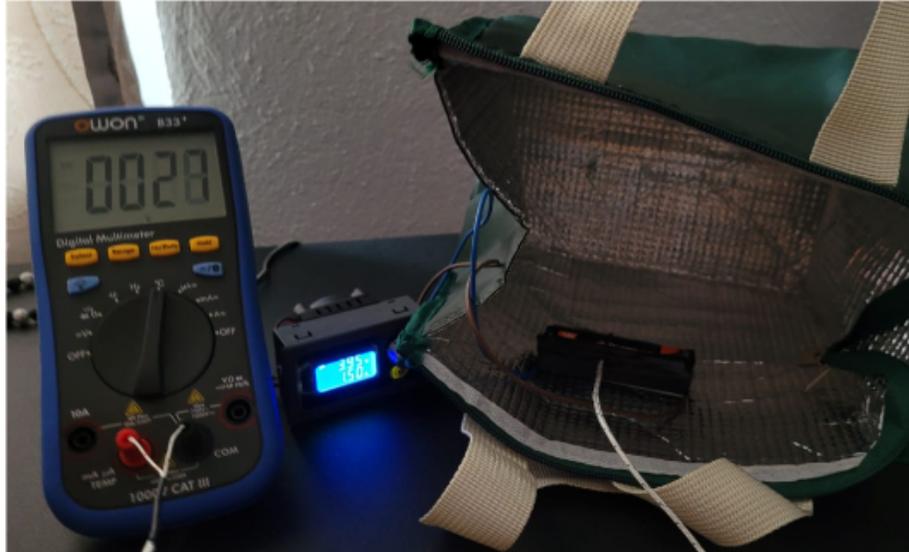


Figura 27: Montaje de todos los elementos necesarios para realizar las pruebas de descarga. Nota: Es necesario cerrar la bolsa térmica antes de iniciar con las pruebas.

Ya que el módulo XY-FZ25 puede medir la descarga en Ah se puede conocer la capacidad de la celda en cada iteración del experimento de descarga, es importante comparar cada intento y verificar que la capacidad medida no presente cambios significativos para confirmar que la celda se encuentra en buenas condiciones.

Por último, para encontrar el voltaje de corte se recomienda tomar mediciones de voltaje en intervalos regulares durante la descarga (por ejemplo, entre 2-5 minutos). Se encontrará el voltaje de corte cuando el nivel de voltaje de la celda empiece a disminuir rápidamente en un periodo de tiempo relativamente corto.

A continuación en la Figura 28 se mostrará un resumen de las mediciones obtenidas en tres diferentes escenarios; en el primero la celda se descarga con una corriente constante de 1.5 A, en el segundo con una corriente de 3 A y por último con una corriente de 4 A.

Celda 3 (1,5 A)				
Tiempo (min)	Temperatura (°C)	mAh (drenado)	Voltaje (V)	% Carga
0	27	0	3,84	100,00%
5	28	129	3,72	86,20%
10	29	255	3,66	72,73%
15	29	380	3,61	59,36%
20	31	508	3,55	45,67%
25	32	632	3,5	32,41%
30	33	756	3,44	19,14%
35	35	882	3,2	5,67%
37	36	935	2,3	0,00%

Celda 3 (3A)				
Tiempo (min)	Temperatura (°C)	mAh (drenado)	Voltaje (V)	% Carga
0	27	0	3,65	100,00%
2	29	108	3,45	88,71%
4	31	215	3,4	77,53%
6	33	317	3,36	66,88%
8	35	419	3,33	56,22%
10	37	519	3,31	45,77%
12	39	621	3,28	35,11%
14	41	722	3,24	24,56%
16	43	823	3,15	14,00%
18	45	922	2,72	3,66%
19	47	957	2,3	0,00%

Celda 3 (4 A)				
Tiempo (min)	Temperatura (°C)	mAh (drenado)	Voltaje (V)	% Carga
0	26	0	3,49	100,00%
2	29	142	3,14	84,45%
4	34	277	3,08	69,86%
6	39	411	3,08	54,98%
8	43	544	3,01	40,42%
10	47	678	2,74	25,74%
12	52	811	2,64	11,17%
14	60	913	2,3	0,00%

Observaciones				
ΔT	9 °C			
Capacidad	935 mAh			
Sin carga Vo	4,07 V			
Vdrop	0,23 V			
Caída rápida Vfast	3,05 V			

Observaciones				
ΔT	20 °C			
Capacidad	957 mAh			
Sin carga Vo	4,1 V			
Vdrop	0,45 V			
Caída rápida Vfast	3,05 V			

Observaciones				
ΔT	34 °C			
Capacidad	913 mAh			
Sin carga Vo	4,1 V			
Vdrop	0,61 V			
Caída rápida Vfast	3,05 V			

Figura 28: Tablas con información obtenida durante 3 distintas pruebas a diferentes tasas de descarga.

De los resultados obtenidos en la Figura 28 se pueden realizar múltiples observaciones:

El voltaje inicial de la celda cuando esta se encuentra bajo demanda va a disminuir dependiendo de la corriente que se esté demandando. Se puede observar que el voltaje inicial en la primera prueba de 1.5 A era de 3.84 V, en la segunda prueba con una demanda de 3 A era de 3.65 V y por último el voltaje inicial en la prueba de 4 A era de 3.49 V.

La capacidad energética de la celda 3 se mantuvo con variaciones mínimas durante la realización de las pruebas, se puede observar que en la primera iteración indica una capacidad de 935 mAh, en la segunda una capacidad de 957 mAh y por último una capacidad de 913 mAh. Por lo tanto, se puede concluir que esta celda está en buen estado y que su capacidad nominal es de 1Ah.

Otra observación que es necesaria realizar es que el voltaje de la celda se mantiene relativamente constante durante el proceso de descarga hasta que el porcentaje de carga disminuye aproximadamente por debajo del 10 %, en ese momento el voltaje comienza a disminuir rápidamente.

Esto se puede notar con facilidad en la segunda prueba de descarga con una tasa de 3 A, se puede observar que el voltaje inicial (100 % de carga) es de 3.65 V, luego de estar

funcionando por alrededor de 16 min y con una carga restante de aproximadamente 14 % el voltaje se lee en 3.15 V, un diferencial de potencial de 0.5 V. Sin embargo, a partir de ese punto solo toma 4 minutos para que el voltaje disminuya rápidamente hasta 2.3 V, una diferencia de voltaje de 0.85 V. Por lo tanto, se puede concluir que el voltaje de corte de esta celda se encuentra aproximadamente a 3.15 V cuando se encuentra bajo una demanda de 3 A.

Por último, se utilizan los valores de temperatura para concluir que el coeficiente C de la celda es aproximadamente 3 C (recordar que esto depende de la capacidad de la batería, si la capacidad de la batería fuera por ejemplo 3 Ah el coeficiente sería 1 C), o en otras palabras 3 A. Ya que se puede observar que bajo esta demanda aún se mantiene en rango de temperatura aceptable, mientras que en las pruebas con una demanda de 4 A la celda alcanza temperaturas de 60° centígrados lo cual ya no se considera una temperatura de operación segura para celdas Li-Ion convencionales. Resumiendo, los datos encontrados son los siguientes:

- Capacidad nominal de 1Ah.
- Coeficiente 3C.
- Voltaje de corte 3.15V (10 % de la carga).

### 8.5.3. Calcular la cantidad de celdas requeridas con base en las características encontradas durante el proceso de caracterización

Nuevamente, para calcular la cantidad de celdas en serie se recomienda que el voltaje final del banco supere el voltaje más alto requerido por el dispositivo, en el caso del Rover UVG este serían los 12 V requeridos por los motores. Por lo tanto:

$$Serie = Voltaje\ requerido / Voltaje\ celda$$

$$Serie = 12\ V / 3.7\ V = 3.24\ celdas$$

Se recomienda redondear hacia arriba, ya que cuando las celdas se descargan o están operando su voltaje disminuye. Por lo tanto:

$$Serie = 4\ celdas$$

Esto a su vez permite conocer el voltaje final del banco:

$$V_{bank} = 4\ celdas * 3.7\ V = 14.8\ V$$

Para calcular la cantidad de celdas que se requieren en paralelo es necesario hacer uso de la capacidad requerida (50 Wh para el Rover UVG) y se procede de la siguiente manera:

$$Paralelo = \frac{Capacidad\ requerida\ en\ Wh}{(Capacidad\ de\ celda\ en\ Ah)(V_{bank})}$$

$$Paralelo = \frac{50\ Wh}{(1\ Ah)(14.8\ V)} = 3.38\ celdas \approx 4\ celdas$$

Por lo tanto, la capacidad final del banco con esta configuración sería de:

$$E = V_{bank} * E_{cell} * N_{paralelo} = (14.8\ V)(3\ Ah) * (2) = 88.8\ Wh$$

$$E = (14.8\ V)(1\ Ah)(4) = 59.2\ Wh$$

En conclusión, el banco de baterías diseñado utilizando las celdas 18650 disponibles en la universidad debe tener una configuración de cuatro celdas en serie y cuatro celdas en paralelo (4S4P), un ejemplo gráfico de las conexiones se puede observar en la Figura 31.

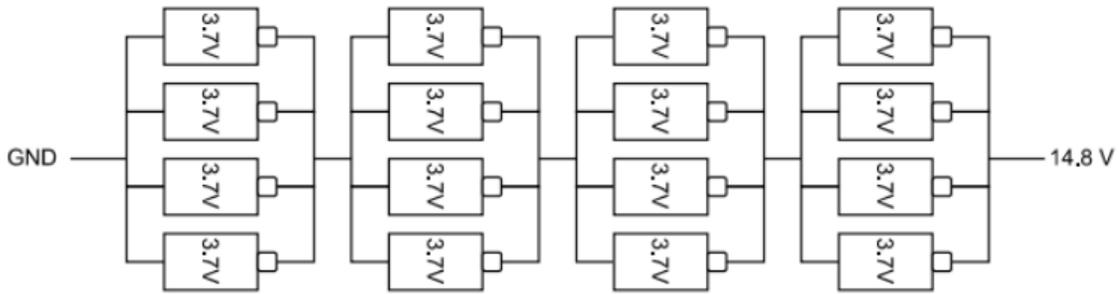


Figura 29: Ejemplo de las conexiones en un banco de baterías con una distribución 4S4P.

#### 8.5.4. Comparativa de costos y tamaños entre los dos bancos de baterías propuestos

Ya que se han planteado dos alternativas para el diseño del banco de baterías se considera pertinente realizar una comparación entre sus capacidades y costos.

El primer banco de baterías diseñado con celdas B30 y cuya distribución puede observarse en la Figura 24 utiliza 5 celdas 18650 en total, de las cuales el costo individual es de Q110.00 brindando un total para el costo del banco de Q550.00.

Mientras que, el segundo banco de baterías utiliza un total de 16 celdas 18650, en el mercado nacional se encontró que el precio de una celda con capacidades similares a las disponibles en el departamento ronda un promedio de Q40.00 lo cual brinda un total de Q640.00 para el costo aproximado del banco.

La comparativa completa se puede observar en la Figura 30.

	Modelo B30	Disponibles en laboratorio	% de diferencia
<b>Costo total</b>	Q 550	Q 640	16%
<b>Volumen aproximado</b>	82.5 cm <sup>3</sup>	264 cm <sup>3</sup>	220%
<b>Capacidad</b>	55.5 Wh	59.2 Wh	7%
<b>Potencia máxima</b>	370 W	178 W	108%

Figura 30: Comparativa de bancos de batería.

En conclusión, se puede observar que el banco de baterías diseñado con las celdas B30 es superior en todos los aspectos listados en el cuadro comparativo de la Figura 30. Esto refleja que al tener la posibilidad de seleccionar a conveniencia qué modelo de celdas se van a utilizar para la construcción del banco de baterías puede representar ventajas significativas.

También es importante destacar que el uso de celdas de alto rendimiento representa ventajas significativas en los tamaños finales de los bancos de baterías a fabricar.

## 8.6. Configuración del integrado

El integrado cuenta con distintos registros de configuración, estos deben ser configurados adecuadamente para que el monitoreo del banco de baterías se realice correctamente y los datos calculados por el integrado sean válidos.

A continuación se muestran todos los registros que deben ser configurados en el integrado BQ76942 para utilizar el banco de baterías diseñado utilizando 5 celdas 18650 modelo B30 en serie:

- *Vcell mode* (0x9304) = 0x001F. Esta configuración especifica al integrado que solamente se conectan 5 celdas.
- *Power config* (0x9234) = 0x2D80. Cambiar los valores del tiempo que se tarda en salir del modo hibernación para evitar *bugs*.
- *TS1 Config* (0x92FD) = 0x07. Configura el termistor 1 para la medir las temperaturas de celdas.
- *HDQ Pin Config* (0x9300) = 0x00. No se utiliza un termistor externo.
- *Enable Protections A* (0x9261) = 0xBC. Activa las protecciones de corto circuito, sobre corriente durante la descarga, sobre corriente durante la carga, sobre voltaje y de bajo voltaje.
- *Enable protections B* (0x9262) = 0xF7. Activa las protecciones de alta temperatura interna, durante la descarga, durante la carga, de baja temperatura interna, durante la carga y durante la descarga.
- *Balancing Configuration* (0x9335) = 0x31. Configura el balanceo de celdas en automático.
- *CUV Threshold* (0x9275) = 0x31. Configura el límite de bajo voltaje en 2479 mV.
- *COV Threshold* (0x9278) = 0x55. configura el límite de alto voltaje en 4301 mV.
- *OCC Threshold* (0x9280) = 0x05. Configura el límite de sobre corriente durante la carga en 10 A asumiendo el uso de una resistencia de 1 mOhm para el muestreo.
- *OCD1 Threshold* (0x9282) = 0x0A. Configura el límite de sobre corriente durante la descarga en 20 A asumiendo el uso de una resistencia de 1 mOhm para el muestreo.
- *SCD Threshold* (0x9286) = 0x05. Configura el límite de corriente para detectar un corto circuito en 100 A asumiendo el uso de una resistencia de 1 mOhm para el muestreo.
- *SCD Delay* (0x9287) = 0x03. Si el límite de corriente para la detección de corto circuito se mantiene por un mínimo de 30 uS entonces se activa la protección por corto circuito.
- *SCDL Latch Limit* (0x9295) = 0x01. Configura el reinicio de operaciones luego de la detección de un corto circuito solamente luego de que se retire la carga.

```

1 void BQ76942::defaultSettings()           //OK
2 {
3
4 // Ingresa el comando CONFIGUPDATE mode (Subcommand 0x0090)
5 // - requerido para modificar la configuración
6 commandOnlySubcommands(SET_CFGUPDATE);
7
8 //POWERCONFIG
9 // Modificar inmediatamente los registros WK_SPD_x en 00 ya que la configuración
10 // por defecto en 01 genera bugs.
11 bqSetRegister(PowerConfig, 0x2D80, 2);
12
13 //Vcell mode
14 // Cambiar a 5 la cantidad de celdas que se van a utilizar
15 bqSetRegister(VCellMode, 0x001F, 2);
16
17 //TS1 config
18 // Configura al termistor 1 para leer la temperatura de las celdas.
19 bqSetRegister(TS1Config, 0x07, 1);
20
21 //HDQ Pin Config
22 // No se utiliza un termistor externo.
23 bqSetRegister(HDQPinConfig, 0x00, 1);
24
25 //PROTECCIONES A
26 // Activa las protecciones: SCD (short-circuit), OCD1 (over-current in discharge), OCC (over-current in charge),
27 // COV (over-voltage), CUV (under-voltage)
28 bqSetRegister(EnabledProtectionsA, 0x8C, 1);
29
30 //PROTECCIONES B
31 // Activa las protecciones de alta temperatura interna durante la descarga y durante la carga,
32 // también de de baja temperatura interna, durante la carga y durante la descarga.
33 bqSetRegister(EnabledProtectionsB, 0xF7, 1);
34
35 //CONFIGURACION DE BALANCEO DE CELDAS
36 // Configura el balanceo de las celdas en automático.
37 bqSetRegister(BalancingConfiguration, 0x31, 1);
38
39 //LIMITE DE BAJO VOLTAJE
40 // Configura el threshold para el bajo voltaje en 2479mV.
41 bqSetRegister(CUVThreshold, 0x31, 1);
42
43 //LIMITE DE ALTO VOLTAJE
44 // Configura el threshold para el alto voltaje en 4301mV.
45 bqSetRegister(COVThreshold, 0x55, 1);
46
47 //LIMITE DE SOBRE CORRIENTE (CARGA)
48 // Configura el límite de sobre corriente durante la carga en 10 A
49 // asumiendo el uso de una resistencia de 1 mOhm para el muestreo.
50 bqSetRegister(OCCThreshold, 0x05, 1);
51
52 //LIMITE DE SOBRE CORRIENTE (DESCARGA)
53 // Configura el límite de sobre corriente durante la descarga en 20 A
54 // asumiendo el uso de una resistencia de 1 mOhm para el muestreo.
55 bqSetRegister(OCD1Threshold, 0x0A, 1);
56
57 //LIMITE DE CORTO CIRCUITO
58 // Configura el límite de corriente para detectar un corto circuito en 100 A
59 // asumiendo el uso de una resistencia de 1 mOhm para el muestreo.
60 bqSetRegister(SCDThreshold, 0x05, 1);
61
62 //DELAY CORTO CIRCUITO
63 // Si el límite de corriente para la detección de corto circuito se mantiene
64 // por un mínimo de 30 uS entonces se activa la protección por corto circuito.
65 bqSetRegister(SCDDelay, 0x03, 1);
66
67 //REANUDAR OPERACIONES
68 // Configura el reinicio de operaciones luego de la detección de un corto
69 // circuito solamente luego de que se retire la carga.
70 bqSetRegister(SCDLLatchLimit, 0x01, 1);
71
72
73 // Salir de CONFIGUPDATE mode - Subcommand 0x0092
74 commandOnlySubcommands(EXIT_CFGUPDATE);
75
76 }

```

Figura 31: Código utilizado para cambiar la configuración del integrado a las requeridas para el banco de baterías diseñado para el Rover UVG.

- Se logró realizar correctamente la lectura de los registros que contienen información sobre el estado del paquete de baterías, por medio de un protocolo de comunicación I2C haciendo uso de un ESP32. Así mismo, se logró realizar correctamente la escritura a los registros de configuración.
- Se desarrolló una librería para el microcontrolador ESP32 que contiene funciones de lectura y escritura tanto para usuarios de nivel avanzado como para usuarios de nivel principiante utilizando el framework de Arduino.
- Se planteó una metodología que sirve como guía para diseñar un paquete de baterías de tecnología Li-ion utilizando el integrado BQ76942 como monitor. Se calculó que el requerimiento energético para el Rover UVG es de 50Wh para lograr una autonomía de 20 min y se validó la metodología propuesta diseñando un banco de baterías con una capacidad de aproximadamente 55 Wh.

Para futuros trabajos de graduación se recomienda:

- Utilizar la metodología propuesta en este trabajo para diseñar e implementar un banco de baterías con distintos requerimientos energéticos, para validar que dicha metodología se adapta a distintas situaciones.
- Modificar la librería desarrollada en este trabajo para que su funcionamiento no dependa del framework de Arduino.
- Desarrollar una nueva librería en lenguaje C en lugar del lenguaje C++ para que se pueda adaptar más fácilmente a otros microcontroladores.
- Implementar el banco propuesto en la validación de la metodología en el proyecto Rover UVG.

- 
- 
- [1] J. Hurtado, “Diseño mecánico y de potencia para el rover de la Universidad del Valle de Guatemala,” UVG, 2022.
  - [2] M. Nizam, H. Maghfiroh, R. A. Rosadi y K. D. U. Kusumaputri, “Battery management system design (BMS) for lithium ion batteries,” AIP, 2020.
  - [3] K.-H. Park, C.-H. Kim, H.-K. Cho y J.-K. Seo, “Design Considerations of a Lithium Ion Battery Management System (BMS) for the STSAT-3 Satellite,” *Journal of Power Electronics*, vol. 10, n.º 2, págs. 210-217, 2010.
  - [4] Z.-Y. Hou, P.-Y. Lou y C.-C. Wang, “State of Charge, State of Health, and State of Function Monitoring for EV BMS,” IEEE, 2017.
  - [5] T. Instruments, *BQ76942EVM*, <https://www.ti.com/tool/BQ76942EVM>, Accedido en mayo de 2023, oct. de 2020.
  - [6] U. of Colorado Boulder, *Algorithms for Battery Management Systems*, <https://www.coursera.org/specializations/algorithms-for-battery-management-systems>, Accedido en mayo de 2023, 2018.
  - [7] T. Instrument, *Battery Management Studio (bqStudio) Software*, <https://www.ti.com/tool/BQSTUDIO>, Accedido en mayo de 2023, 2016.
  - [8] M. Sunna, “User’s Guide BQ76942 Evaluation Module,” Texas Instruments, 2020.
  - [9] Espressif, *Arduino-Esp32*, <https://docs.espressif.com/projects/arduino-esp32/en/latest/libraries.html>, Accedido en septiembre de 2023.
  - [10] *How to read a register value using the Wire library*, <https://forum.arduino.cc/t/how-to-read-a-register-value-using-the-wire-library/206123/3>, Accedido en septiembre de 2023, 2014.
  - [11] “BQ76942 Technical Reference Manual,” Texas Instruments, 2022.
  - [12] T. Instruments, *BQ76942*, <https://www.ti.com/product/BQ76942>, Accedido en mayo de 2023, 2020.

- [13] T. Instruments, *Microcontroller programming for the BQ769x2 Battery Monitor Family*, [https://youtu.be/RfyzTKTLQCM?si=OPiK\\_g-gyjUvDCa](https://youtu.be/RfyzTKTLQCM?si=OPiK_g-gyjUvDCa), Accedido en septiembre de 2023, 2022.
- [14] Jay, *How do you write to a specific register using I2C communication?* <https://arduino.stackexchange.com/questions/68449/how-do-you-write-to-a-specific-register-using-i2c-communication>, Accedido en septiembre de 2023, 2019.
- [15] T. R. Kuphaldt, "Lessons in electric circuit volume I - DC," 2006.
- [16] Ada y D. Nosenowitz, *Power Capacity and Power Capability*, <https://learn.adafruit.com/all-about-batteries/power-capacity-and-power-capability>, Accedido en septiembre de 2023, feb. de 2013.
- [17] G. Vapor, *BATERÍA LI-ION 18650 BAK N18650CNP DE 2,500 MAH 12C*, <https://guatevapor.com/product/vapcell-bateria-18650-3000-mah/>, Accedido en septiembre de 2023.
- [18] B. Sacks, *Building A Lithium Battery Pack From 18650 Cells*, Accedido en septiembre de 2023, sep. de 2023.
- [19] Wissblue, *wissblue LCD Display 18650 Battery Charger*, <https://wissbluearmory.com/products/wb-wissblue-lcd-display18650-battery-charger-2a-fast-charger-for-li-ion-batteries-26650-21700-18490-18350-17670-17500-16340-14500-rcr123a-ni-mh-ni-cd-a-aa-aaa>, Accedido en noviembre de 2023.
- [20] L. electronica, *CARGA ELECTRÓNICA AJUSTABLE XY-FZ25 DE 25W 4A*, <https://laelectronica.com.gt/carga-electronica-ajustable-xy-fz25-de-25w-4a>, Accedido en noviembre de 2023.

### 12.1. Repositorio a la librería para el manejo del integrado BQ76942

- <https://github.com/mtrugt/BQ769x2>

### 12.2. Tablas con direcciones de registros

## 12.1 Direct Commands

**Table 12-1. Direct Commands Table**

Command	Name	Units	Type	Access	Description
0x00	Control Status	Hex	H2	Sealed: R/W Unsealed: R/W Full Access: R/W	When read, this command provides device status bits. This command behaves similarly to 0x3E/0x3F when written. When read back immediately after word write, it will return 0xFFA5 once. Subsequent reads will return Control Status. Writing this command is used for legacy auto-detection, and it is not recommended for customers to write to it. Bit descriptions can be found in <a href="#">Control Status Register</a> .
0x02	Safety Alert A	Hex	H1	Sealed: R Unsealed: R Full Access: R	Provides individual alert signals when enabled safety alerts have triggered. Bit descriptions can be found in <a href="#">Safety Alert A Register</a> .
0x03	Safety Status A	Hex	H1	Sealed: R Unsealed: R Full Access: R	Provides individual fault signals when enabled safety faults have triggered. Bit descriptions can be found in <a href="#">Safety Status A Register</a> .
0x04	Safety Alert B	Hex	H1	Sealed: R Unsealed: R Full Access: R	Provides individual alert signals when enabled safety alerts have triggered. Bit descriptions can be found in <a href="#">Safety Alert B Register</a> .
0x05	Safety Status B	Hex	H1	Sealed: R Unsealed: R Full Access: R	Provides individual fault signals when enabled safety faults have triggered. Bit descriptions can be found in <a href="#">Safety Status B Register</a> .
0x06	Safety Alert C	Hex	H1	Sealed: R Unsealed: R Full Access: R	Provides individual alert signals when enabled safety alerts have triggered. Bit descriptions can be found in <a href="#">Safety Alert C Register</a> .
0x07	Safety Status C	Hex	H1	Sealed: R Unsealed: R Full Access: R	Provides individual fault signals when enabled safety faults have triggered. Bit descriptions can be found in <a href="#">Safety Status C Register</a> .
0x0A	PF Alert A	Hex	H1	Sealed: R Unsealed: R Full Access: R	Provides individual alert signals when enabled Permanent Fail alerts have triggered. Bit descriptions can be found in <a href="#">PF Alert A Register</a> .
0x0B	PF Status A	Hex	H1	Sealed: R Unsealed: R Full Access: R	Provides individual fault signals when enabled Permanent Fail faults have triggered. Bit descriptions can be found in <a href="#">PF Status A Register</a> .
0x0C	PF Alert B	Hex	H1	Sealed: R Unsealed: R Full Access: R	Provides individual alert signals when enabled Permanent Fail alerts have triggered. Bit descriptions can be found in <a href="#">PF Alert B Register</a> .
0x0D	PF Status B	Hex	H1	Sealed: R Unsealed: R Full Access: R	Provides individual fault signals when enabled Permanent Fail faults have triggered. Bit descriptions can be found in <a href="#">PF Status B Register</a> .
0x0E	PF Alert C	Hex	H1	Sealed: R Unsealed: R Full Access: R	Provides individual alert signals when enabled Permanent Fail alerts have triggered. Bit descriptions can be found in <a href="#">PF Alert C Register</a> .
0x0F	PF Status C	Hex	H1	Sealed: R Unsealed: R Full Access: R	Provides individual fault signals when enabled Permanent Fail faults have triggered. Bit descriptions can be found in <a href="#">PF Status C Register</a> .
0x10	PF Alert D	Hex	H1	Sealed: R Unsealed: R Full Access: R	Provides individual alert signals when enabled Permanent Fail alerts have triggered. Bit descriptions can be found in <a href="#">PF Alert D Register</a> .

**Table 12-1. Direct Commands Table (continued)**

Command	Name	Units	Type	Access	Description
0x11	PF Status D	Hex	H1	Sealed: R Unsealed: R Full Access: R	Provides individual fault signals when enabled Permanent Fail faults have triggered. Bit descriptions can be found in <a href="#">PF Status D Register</a> .
0x12	Battery Status	Hex	H2	Sealed: R Unsealed: R Full Access: R	Flags related to battery status Bit descriptions can be found in <a href="#">Battery Status Register</a> .
0x14	Cell 1 Voltage	mV	I2	Sealed: R Unsealed: R Full Access: R	16-bit voltage on cell 1
0x16	Cell 2 Voltage	mV	I2	Sealed: R Unsealed: R Full Access: R	16-bit voltage on cell 2
0x18	Cell 3 Voltage	mV	I2	Sealed: R Unsealed: R Full Access: R	16-bit voltage on cell 3
0x1A	Cell 4 Voltage	mV	I2	Sealed: R Unsealed: R Full Access: R	16-bit voltage on cell 4
0x1C	Cell 5 Voltage	mV	I2	Sealed: R Unsealed: R Full Access: R	16-bit voltage on cell 5
0x1E	Cell 6 Voltage	mV	I2	Sealed: R Unsealed: R Full Access: R	16-bit voltage on cell 6
0x20	Cell 7 Voltage	mV	I2	Sealed: R Unsealed: R Full Access: R	16-bit voltage on cell 7
0x22	Cell 8 Voltage	mV	I2	Sealed: R Unsealed: R Full Access: R	16-bit voltage on cell 8
0x24	Cell 9 Voltage	mV	I2	Sealed: R Unsealed: R Full Access: R	16-bit voltage on cell 9
0x26	Cell 10 Voltage	mV	I2	Sealed: R Unsealed: R Full Access: R	16-bit voltage on cell 10
0x34	Stack Voltage	userV	I2	Sealed: R Unsealed: R Full Access: R	16-bit voltage on top of stack
0x36	PACK Pin Voltage	userV	I2	Sealed: R Unsealed: R Full Access: R	16-bit voltage on PACK pin
0x38	LD Pin Voltage	userV	I2	Sealed: R Unsealed: R Full Access: R	16-bit voltage on LD pin
0x3A	CC2 Current	userA	I2	Sealed: R Unsealed: R Full Access: R	16-bit CC2 current
0x62	Alarm Status	Hex	H2	Sealed: R/W Unsealed: R/W Full Access: R/W	Latched signal used to assert the ALERT pin. Write a bit high to clear the latch. Bit descriptions can be found in <a href="#">Alarm Status Register</a> .
0x64	Alarm Raw Status	Hex	H2	Sealed: R Unsealed: R Full Access: R	Unlatched value of flags which can be selected to be latched (using <i>Alarm Enable()</i> ) and used to assert the ALERT pin. Bit descriptions can be found in <a href="#">Alarm Raw Status Register</a> .
0x66	Alarm Enable	Hex	H2	Sealed: R/W Unsealed: R/W Full Access: R/W	Mask for <i>Alarm Status()</i> . Can be written to change during operation to change which alarm sources are enabled. The default value of this parameter is set by <b>Settings:Alarm:Default Alarm Mask</b> . Bit descriptions can be found in <a href="#">Alarm Enable Register</a> .
0x68	Int Temperature	0.1 K	I2	Sealed: R Unsealed: R Full Access: R	This is the most recent measured internal die temperature.

**Table 12-22. Table of Command-Only Subcommands**

Command	Name	Access	Description
0x000E	EXIT_DEEPSLEEP	Sealed: W Unsealed: W Full Access: W	Exit DEEPSLEEP mode
0x000F	DEEPSLEEP	Sealed: W Unsealed: W Full Access: W	Enter DEEPSLEEP mode. Must be sent twice in a row within 4 s to take effect
0x0010	SHUTDOWN	Sealed: W Unsealed: W Full Access: W	Start SHUTDOWN sequence. Must be sent twice in a row within 4 s to take effect if sealed. If sent twice while unsealed, the shutdown delays are skipped.
0x0012	RESET	Sealed: — Unsealed: W Full Access: W	Reset the device
0x001C	PDSGTEST	Sealed: — Unsealed: W Full Access: W	In FET Test mode, toggle the PDSG FET enable
0x001D	FUSE_TOGGLE	Sealed: — Unsealed: W Full Access: W	Toggle FUSE state
0x001E	PCHGTEST	Sealed: — Unsealed: W Full Access: W	In FET Test mode, toggle the PCHG FET enable
0x001F	CHGTEST	Sealed: — Unsealed: W Full Access: W	In FET Test mode, toggle the CHG FET enable
0x0020	DSGTEST	Sealed: — Unsealed: W Full Access: W	In FET Test mode, toggle the DSG FET enable
0x0022	FET_ENABLE	Sealed: — Unsealed: W Full Access: W	Toggle FET_EN in Manufacturing Status. FET_EN = 0 means FET Test Mode. FET_EN = 1 means Firmware FET Control
0x0024	PF_ENABLE	Sealed: — Unsealed: W Full Access: W	Toggle PF_EN in Manufacturing Status

**Table 12-23. Subcommands Table**

Command	Name	Access	Offset	Data	Units	Type	Description
0x0001	DEVICE_NUMBER	Sealed: R Unsealed: R Full Access: R	0	Device Number	Hex	U2	Reports the device number that identifies the product. The data is returned in little-endian format.
0x0002	FW_VERSION	Sealed: R Unsealed: R Full Access: R	0	Device Number (Big-Endian)	Hex	U2	Device number in big-endian format for compatibility with legacy products
			2	Firmware Version (Big-Endian)	Hex	U2	Device firmware major and minor version number (Big-Endian)
			4	Build Number (Big-Endian)	Hex	U2	Firmware build number in big-endian, binary-coded decimal format for compatibility with legacy products
0x0003	HW_VERSION	Sealed: R Unsealed: R Full Access: R	0	Hardware Version	Hex	U2	Reports the device hardware version number.
0x0004	IROM_SIG	Sealed: R Unsealed: R Full Access: R	0	Instruction ROM Signature	Hex	U2	Calculates and reports the device instruction ROM signature.
0x0005	STATIC_CFG_SIG	Sealed: R Unsealed: R Full Access: R	0	Static Configuration Signature	Hex	U2	The lower 15-bits report the signature of static (non-calibration) configuration data memory. If this does not match the stored <b>System Data: Integrity: Config RAM Signature</b> , the MSBit is set.

**Table 12-23. Subcommands Table (continued)**

Command	Name	Access	Offset	Data	Units	Type	Description
0x0097	FET_CONTROL	Sealed: W Unsealed: W Full Access: W	0	FET Control	Hex	H1	Allows host control of individual FET drivers. This subcommand should not be used if the DDSG or DCHG pin is being used in DDSG/DCHG mode. Bit descriptions can be found in <a href="#">FET Control Register</a> .
0x0098	REG12_CONTROL	Sealed: W Unsealed: W Full Access: W	0	REG12 Control	Hex	H1	Changes voltage regulator settings. Bit descriptions can be found in <a href="#">REG12 Control Register</a> .
0x00a0	OTP_WR_CHECK	Sealed: — Unsealed: R Full Access: R	0	OTP Write Check Result	Hex	H1	Reports whether or not OTP programming is allowed. Bit descriptions can be found in <a href="#">OTP Write Check Result Register</a> .
			1	OTP Write Check Data Fail Addr	Hex	U2	When data cannot be programmed to OTP because no XOR bits remain, this will contain the address of the first data value which could not be programmed.
0x00a1	OTP_WRITE	Sealed: — Unsealed: R Full Access: R	0	OTP Write Result	Hex	H1	Reports whether or not OTP programming is allowed. Bit descriptions can be found in <a href="#">OTP Write Result Register</a> .
			1	OTP Write Data Fail Addr	Hex	U2	When data cannot be programmed to OTP because no XOR bits remain, this will contain the address of the first data value which could not be programmed.
0xf081	READ_CAL1	Sealed: — Unsealed: R Full Access: R	0	Calibration Data Counter	—	I2	Sample counter that is incremented when buffer is updated. Used to ensure unique measurement samples are taken when averaging is required.
			2	CC2 Counts	—	I4	32-bit CC2 Counts from the most recent current measurement.
			6	PACK pin ADC Counts	—	I2	16-bit ADC counts from previous PACK pin voltage measurement.
			8	Top of Stack ADC Counts	—	I2	16-bit ADC counts from previous Top of Stack voltage measurement.
			10	LD pin ADC Counts	—	I2	16-bit ADC counts from previous Top of Stack voltage measurement.
0xf090	CAL_CUV	Sealed: — Unsealed: R Full Access: R	0	CUV Threshold Override	Hex	U2	Calibrates CUV using the top cell input to set <b>Calibration:COV:COV Threshold Override</b> . By using this calibration command, threshold levels between the normal 50-mV steps can be set. Only available in CONFIG_UPDATE mode.
0xf091	CAL_COV	Sealed: — Unsealed: R Full Access: R	0	COV Threshold Override	Hex	U2	Calibrates COV using the top cell input to set <b>Calibration:COV:COV Threshold Override</b> . By using this calibration command, threshold levels between the normal 50-mV steps can be set. Only available in CONFIG_UPDATE mode.

**Table 13-37. Data Memory Table (continued)**

Class	Subclass	Address	Name	Type	Min Value	Max Value	Default	Units
Calibration	180K Temperature Model	0x920E	Coeff b3	I2	-32768	32767	-3427	—
Calibration	180K Temperature Model	0x9210	Coeff b4	I2	-32768	32767	4385	—
Calibration	180K Temperature Model	0x9214	Adc0	I2	-32768	32767	17246	—
Calibration	Custom Temperature Model	0x9216	Coeff a1	I2	-32768	32767	0	—
Calibration	Custom Temperature Model	0x9218	Coeff a2	I2	-32768	32767	0	—
Calibration	Custom Temperature Model	0x921A	Coeff a3	I2	-32768	32767	0	—
Calibration	Custom Temperature Model	0x921C	Coeff a4	I2	-32768	32767	0	—
Calibration	Custom Temperature Model	0x921E	Coeff a5	I2	-32768	32767	0	—
Calibration	Custom Temperature Model	0x9220	Coeff b1	I2	-32768	32767	0	—
Calibration	Custom Temperature Model	0x9222	Coeff b2	I2	-32768	32767	0	—
Calibration	Custom Temperature Model	0x9224	Coeff b3	I2	-32768	32767	0	—
Calibration	Custom Temperature Model	0x9226	Coeff b4	I2	-32768	32767	0	—
Calibration	Custom Temperature Model	0x9228	Rc0	I2	-32768	32767	0	—
Calibration	Custom Temperature Model	0x922A	Adc0	I2	-32768	32767	0	—
Calibration	Current Deadband	0x922D	Coulomb Counter Deadband	U1	0	255	9	234nV
Calibration	CUV	0x91D4	CUV Threshold Override	U2	0x0000	0xFFFF	0xFFFF	Hex
Calibration	COV	0x91D6	COV Threshold Override	U2	0x0000	0xFFFF	0xFFFF	Hex
Settings	Fuse	0x9231	Min Blow Fuse Voltage	I2	0	32767	500	10mV
Settings	Fuse	0x9233	Fuse Blow Timeout	U1	0	255	30	s
Settings	Configuration	0x9234	Power Config	H2	0x0000	0xFFFF	0x2982	Hex
Settings	Configuration	0x9236	REG12 Config	H1	0x00	0xFF	0x00	Hex
Settings	Configuration	0x9237	REG0 Config	H1	0x00	0x03	0x00	Hex
Settings	Configuration	0x9238	HWD Regulator Options	H1	0x00	0xFF	0x00	Hex
Settings	Configuration	0x9239	Comm Type	U1	0x00	0x1F	0	—
Settings	Configuration	0x923A	I2C Address	U1	0x00	0xFF	0	—
Settings	Configuration	0x923C	SPI Configuration	H1	0x00	0x7F	0x20	—
Settings	Configuration	0x923D	Comm Idle Time	U1	0	255	0	s
Settings	Configuration	0x92FA	CFETOFF Pin Config	H1	0x00	0xFF	0x00	Hex
Settings	Configuration	0x92FB	DFETOFF Pin Config	H1	0x00	0xFF	0x00	Hex
Settings	Configuration	0x92FC	ALERT Pin Config	H1	0x00	0xFF	0x00	Hex
Settings	Configuration	0x92FD	TS1 Config	H1	0x00	0xFF	0x07	Hex
Settings	Configuration	0x92FE	TS2 Config	H1	0x00	0xFF	0x00	Hex
Settings	Configuration	0x92FF	TS3 Config	H1	0x00	0xFF	0x00	Hex
Settings	Configuration	0x9300	HDQ Pin Config	H1	0x00	0xFF	0x00	Hex
Settings	Configuration	0x9301	DCHG Pin Config	H1	0x00	0xFF	0x00	Hex
Settings	Configuration	0x9302	DDSG Pin Config	H1	0x00	0xFF	0x00	Hex
Settings	Configuration	0x9303	DA Configuration	H1	0x00	0xFF	0x05	Hex
Settings	Configuration	0x9304	Vcell Mode	H2	0x0000	0x03FF	0x0000	Hex
Settings	Configuration	0x9307	CC3 Samples	U1	2	255	80	Num
Settings	Protection	0x925F	Protection Configuration	H2	0x0000	0x07FF	0x0002	Hex

**Table 13-37. Data Memory Table (continued)**

Class	Subclass	Address	Name	Type	Min Value	Max Value	Default	Units
Settings	Protection	0x9261	Enabled Protections A	U1	0x00	0xFF	0x88	Hex
Settings	Protection	0x9262	Enabled Protections B	U1	0x00	0xFF	0x00	Hex
Settings	Protection	0x9263	Enabled Protections C	U1	0x00	0xFF	0x00	Hex
Settings	Protection	0x9265	CHG FET Protections A	U1	0x00	0xFF	0x98	Hex
Settings	Protection	0x9266	CHG FET Protections B	U1	0x00	0xFF	0xD5	Hex
Settings	Protection	0x9267	CHG FET Protections C	U1	0x00	0xFF	0x56	Hex
Settings	Protection	0x9269	DSG FET Protections A	U1	0x00	0xFF	0xE4	Hex
Settings	Protection	0x926A	DSG FET Protections B	U1	0x00	0xFF	0xE6	Hex
Settings	Protection	0x926B	DSG FET Protections C	U1	0x00	0xFF	0xE2	Hex
Settings	Protection	0x9273	Body Diode Threshold	I2	0	32767	50	mA
Settings	Alarm	0x926D	Default Alarm Mask	H2	0x0000	0xFFFF	0xF800	Hex
Settings	Alarm	0x926F	SF Alert Mask A	U1	0x00	0xFF	0xFC	Hex
Settings	Alarm	0x9270	SF Alert Mask B	U1	0x00	0xFF	0xF7	Hex
Settings	Alarm	0x9271	SF Alert Mask C	U1	0x00	0xFF	0xF4	Hex
Settings	Alarm	0x92C4	PF Alert Mask A	U1	0x00	0xFF	0x5F	Hex
Settings	Alarm	0x92C5	PF Alert Mask B	U1	0x00	0xFF	0x9F	Hex
Settings	Alarm	0x92C6	PF Alert Mask C	U1	0x00	0xFF	0x00	Hex
Settings	Alarm	0x92C7	PF Alert Mask D	U1	0x00	0xFF	0x00	Hex
Settings	Permanent Failure	0x92C0	Enabled PF A	U1	0x00	0xFF	0x00	Hex
Settings	Permanent Failure	0x92C1	Enabled PF B	U1	0x00	0xFF	0x00	Hex
Settings	Permanent Failure	0x92C2	Enabled PF C	U1	0x00	0xFF	0x07	Hex
Settings	Permanent Failure	0x92C3	Enabled PF D	U1	0x00	0xFF	0x00	Hex
Settings	FET	0x9308	FET Options	H1	0x00	0xFF	0x0D	Hex
Settings	FET	0x9309	Chg Pump Control	U1	0x00	0xFF	0x01	Hex
Settings	FET	0x930A	Precharge Start Voltage	I2	0	32767	0	mV
Settings	FET	0x930C	Precharge Stop Voltage	I2	0	32767	0	mV
Settings	FET	0x930E	Predischarge Timeout	U1	0	255	5	10ms
Settings	FET	0x930F	Predischarge Stop Delta	U1	0	255	50	10mV
Settings	Current Thresholds	0x9310	Dsg Current Threshold	I2	0	32767	100	userA
Settings	Current Thresholds	0x9312	Chg Current Threshold	I2	0	32767	50	userA
Settings	Cell Open-Wire	0x9314	Check Time	U1	0	255	5	s
Settings	Interconnect Resistances	0x9315	Cell 1 Interconnect	I2	0	32767	0	mΩ
Settings	Interconnect Resistances	0x9317	Cell 2 Interconnect	I2	0	32767	0	mΩ
Settings	Interconnect Resistances	0x9319	Cell 3 Interconnect	I2	0	32767	0	mΩ
Settings	Interconnect Resistances	0x931B	Cell 4 Interconnect	I2	0	32767	0	mΩ
Settings	Interconnect Resistances	0x931D	Cell 5 Interconnect	I2	0	32767	0	mΩ
Settings	Interconnect Resistances	0x931F	Cell 6 Interconnect	I2	0	32767	0	mΩ
Settings	Interconnect Resistances	0x9321	Cell 7 Interconnect	I2	0	32767	0	mΩ
Settings	Interconnect Resistances	0x9323	Cell 8 Interconnect	I2	0	32767	0	mΩ
Settings	Interconnect Resistances	0x9325	Cell 9 Interconnect	I2	0	32767	0	mΩ
Settings	Interconnect Resistances	0x9327	Cell 10 Interconnect	I2	0	32767	0	mΩ
Settings	Manufacturing	0x9343	Mfg Status Init	H2	0x0000	0xFFFF	0x0040	Hex

**Table 13-37. Data Memory Table (continued)**

Class	Subclass	Address	Name	Type	Min Value	Max Value	Default	Units
Settings	Cell Balancing Config	0x9335	Balancing Configuration	H1	0x00	0xFF	0x00	Hex
Settings	Cell Balancing Config	0x9336	Min Cell Temp	I1	-128	127	-20	°C
Settings	Cell Balancing Config	0x9337	Max Cell Temp	I1	-128	127	60	°C
Settings	Cell Balancing Config	0x9338	Max Internal Temp	I1	-128	127	70	°C
Settings	Cell Balancing Config	0x9339	Cell Balance Interval	U1	1	255	20	s
Settings	Cell Balancing Config	0x933A	Cell Balance Max Cells	U1	0	16	1	Num
Settings	Cell Balancing Config	0x933B	Cell Balance Min Cell V (Charge)	I2	0	5000	3900	mV
Settings	Cell Balancing Config	0x933D	Cell Balance Min Delta (Charge)	U1	0	255	40	mV
Settings	Cell Balancing Config	0x933E	Cell Balance Stop Delta (Charge)	U1	0	255	20	mV
Settings	Cell Balancing Config	0x933F	Cell Balance Min Cell V (Relax)	I2	0	5000	3900	mV
Settings	Cell Balancing Config	0x9341	Cell Balance Min Delta (Relax)	U1	0	255	40	mV
Settings	Cell Balancing Config	0x9342	Cell Balance Stop Delta (Relax)	U1	0	255	20	mV
Power	Shutdown	0x923F	Shutdown Cell Voltage	I2	0	32767	0	mV
Power	Shutdown	0x9241	Shutdown Stack Voltage	I2	0	32767	600	10mV
Power	Shutdown	0x9243	Low V Shutdown Delay	U1	0	63	1	s
Power	Shutdown	0x9244	Shutdown Temperature	U1	0	150	85	°C
Power	Shutdown	0x9245	Shutdown Temperature Delay	U1	0	254	5	s
Power	Shutdown	0x9252	FET Off Delay	U1	0	127	0	0.25s
Power	Shutdown	0x9253	Shutdown Command Delay	U1	0	254	0	0.25s
Power	Shutdown	0x9254	Auto Shutdown Time	U1	0	250	0	min
Power	Shutdown	0x9255	RAM Fail Shutdown Time	U1	0	255	5	s
Power	Sleep	0x9248	Sleep Current	I2	0	32767	20	mA
Power	Sleep	0x924A	Voltage Time	U1	1	255	5	s
Power	Sleep	0x924B	Wake Comparator Current	I2	500	32767	500	mA
Power	Sleep	0x924D	Sleep Hysteresis Time	U1	0	255	10	s
Power	Sleep	0x924E	Sleep Charger Voltage Threshold	I2	0	32767	2000	10mV
Power	Sleep	0x9250	Sleep Charger PACK-TOS Delta	I2	10	8500	200	10mV
System Data	Integrity	0x91E0	Config RAM Signature	U2	0x0000	0x7FFF	0	Hex
Protections	CUV	0x9275	Threshold	U1	20	90	50	50.6mV
Protections	CUV	0x9276	Delay	U2	1	2047	74	3.3 ms
Protections	CUV	0x927B	Recovery Hysteresis	U1	2	20	2	50.6mV
Protections	COV	0x9278	Threshold	U1	20	110	86	50.6mV
Protections	COV	0x9279	Delay	U2	1	2047	74	3.3 ms
Protections	COV	0x927C	Recovery Hysteresis	U1	2	20	2	50.6mV
Protections	COVL	0x927D	Latch Limit	U1	0	255	0	—
Protections	COVL	0x927E	Counter Dec Delay	U1	0	255	10	s
Protections	COVL	0x927F	Recovery Time	U1	0	255	15	s
Protections	OCC	0x9280	Threshold	U1	2	62	2	2mV
Protections	OCC	0x9281	Delay	U1	1	127	4	3.3 ms
Protections	OCC	0x9288	Recovery Threshold	I2	-32768	32767	-200	mA

**Table 13-37. Data Memory Table (continued)**

Class	Subclass	Address	Name	Type	Min Value	Max Value	Default	Units
Protections	OCC	0x92B0	PACK-TOS Delta	I2	10	8500	200	10mV
Protections	OCD1	0x9282	Threshold	U1	2	100	4	2mV
Protections	OCD1	0x9283	Delay	U1	1	127	1	3.3 ms
Protections	OCD2	0x9284	Threshold	U1	2	100	3	2mV
Protections	OCD2	0x9285	Delay	U1	1	127	7	3.3 ms
Protections	SCD	0x9286	Threshold	U1	0	15	0	—
Protections	SCD	0x9287	Delay	U1	1	31	2	15µs
Protections	SCD	0x9294	Recovery Time	U1	0	255	5	s
Protections	OCD3	0x928A	Threshold	I2	-32768	0	-4000	userA
Protections	OCD3	0x928C	Delay	U1	0	255	2	s
Protections	OCD	0x928D	Recovery Threshold	I2	-32768	32767	200	mA
Protections	OCDL	0x928F	Latch Limit	U1	0	255	0	—
Protections	OCDL	0x9290	Counter Dec Delay	U1	0	255	10	s
Protections	OCDL	0x9291	Recovery Time	U1	0	255	15	s
Protections	OCDL	0x9292	Recovery Threshold	I2	-32768	32767	200	mA
Protections	SCDL	0x9295	Latch Limit	U1	0	255	0	—
Protections	SCDL	0x9296	Counter Dec Delay	U1	0	255	10	s
Protections	SCDL	0x9297	Recovery Time	U1	0	255	15	s
Protections	SCDL	0x9298	Recovery Threshold	I2	-32768	32767	200	mA
Protections	OTC	0x929A	Threshold	I1	-40	120	55	°C
Protections	OTC	0x929B	Delay	U1	0	255	2	s
Protections	OTC	0x929C	Recovery	I1	-40	120	50	°C
Protections	OTD	0x929D	Threshold	I1	-40	120	60	°C
Protections	OTD	0x929E	Delay	U1	0	255	2	s
Protections	OTD	0x929F	Recovery	I1	-40	120	55	°C
Protections	OTF	0x92A0	Threshold	U1	0	150	80	°C
Protections	OTF	0x92A1	Delay	U1	0	255	2	s
Protections	OTF	0x92A2	Recovery	U1	0	150	65	°C
Protections	OTINT	0x92A3	Threshold	I1	-40	120	85	°C
Protections	OTINT	0x92A4	Delay	U1	0	255	2	s
Protections	OTINT	0x92A5	Recovery	I1	-40	120	80	°C
Protections	UTC	0x92A6	Threshold	I1	-40	120	0	°C
Protections	UTC	0x92A7	Delay	U1	0	255	2	s
Protections	UTC	0x92A8	Recovery	I1	-40	120	5	°C
Protections	UTD	0x92A9	Threshold	I1	-40	120	0	°C
Protections	UTD	0x92AA	Delay	U1	0	255	2	s
Protections	UTD	0x92AB	Recovery	I1	-40	120	5	°C
Protections	UTINT	0x92AC	Threshold	I1	-40	120	-20	°C
Protections	UTINT	0x92AD	Delay	U1	0	255	2	s
Protections	UTINT	0x92AE	Recovery	I1	-40	120	-15	°C
Protections	Recovery	0x92AF	Time	U1	0	255	3	s
Protections	HWD	0x92B2	Delay	U2	0	65535	60	s
Protections	Load Detect	0x92B4	Active Time	U1	0	255	0	s
Protections	Load Detect	0x92B5	Retry Delay	U1	0	255	50	s
Protections	Load Detect	0x92B6	Timeout	U2	0	65535	1	hrs
Protections	PTO	0x92BA	Charge Threshold	I2	-32768	32767	250	mA
Protections	PTO	0x92BC	Delay	U2	0	65535	1800	s
Protections	PTO	0x92BE	Reset	I2	0	10000	2	userAh
Permanent Fail	CUDEP	0x92C8	Threshold	I2	0	32767	1500	mV
Permanent Fail	CUDEP	0x92CA	Delay	U1	0	255	2	s

