Multi-Cell Li-Ion Battery Management System Using MSP430F5529 and bq76PL536

Daniel Torres *MSP430 System Solutions*

ABSTRACT

This application report explains the implementation of a multi-cell lithium-ion battery management system using an MSP430™ microcontroller and the bq76PL536. The battery manager is implemented using the standard evaluation boards for the MSP430 MCU and the bq76PL536.

The bq76PL536 can be stacked vertically to monitor up to 192 cells without additional isolation components between ICs. A high-speed serial peripheral interface (SPI) bus operates between each bq76PL536 and the MSP430 microcontroller to provide reliable communications through a high-voltage battery cell stack. The battery management system can communicate with an external host or battery charger using USB communication or asynchronous serial communication such as RS232 or RS485.

This application report demonstrates the following features: passive cell balancing, cell undervoltage monitor, cell overvoltage monitor, safety cell overvoltage monitor, cell overtemperature monitor. It also detects whether the battery system is in charge or discharge mode by detecting changes of cell voltages.

Sample application code and other information associated with this application report can be downloaded from http://www.ti.com/lit/zip/slaa478.

NOTE: This application note is applicable to all the MSP430 devices, the source code provided with this document can be used as is with the MSP430F5xx family.

Contents

List of Figures

List of Tables

MSP430 is a trademark of Texas Instruments.

1 Hardware

The battery management system implemented in this application report is based on the existing evaluation modules for the MSP430F5529 and the bq76PL536. Figure 1 shows the system block diagram. For more information on these devices, see the device data sheets.[1][2]

The evaluation modules' part numbers are MSP-TS430PN80USB and BQ76PL536EVM-3. These boards are available from the TI eStore (https://estore.ti.com/). For more details and information related to these evaluation modules (EVMs), see the specific EVM user's guide.[3][4]

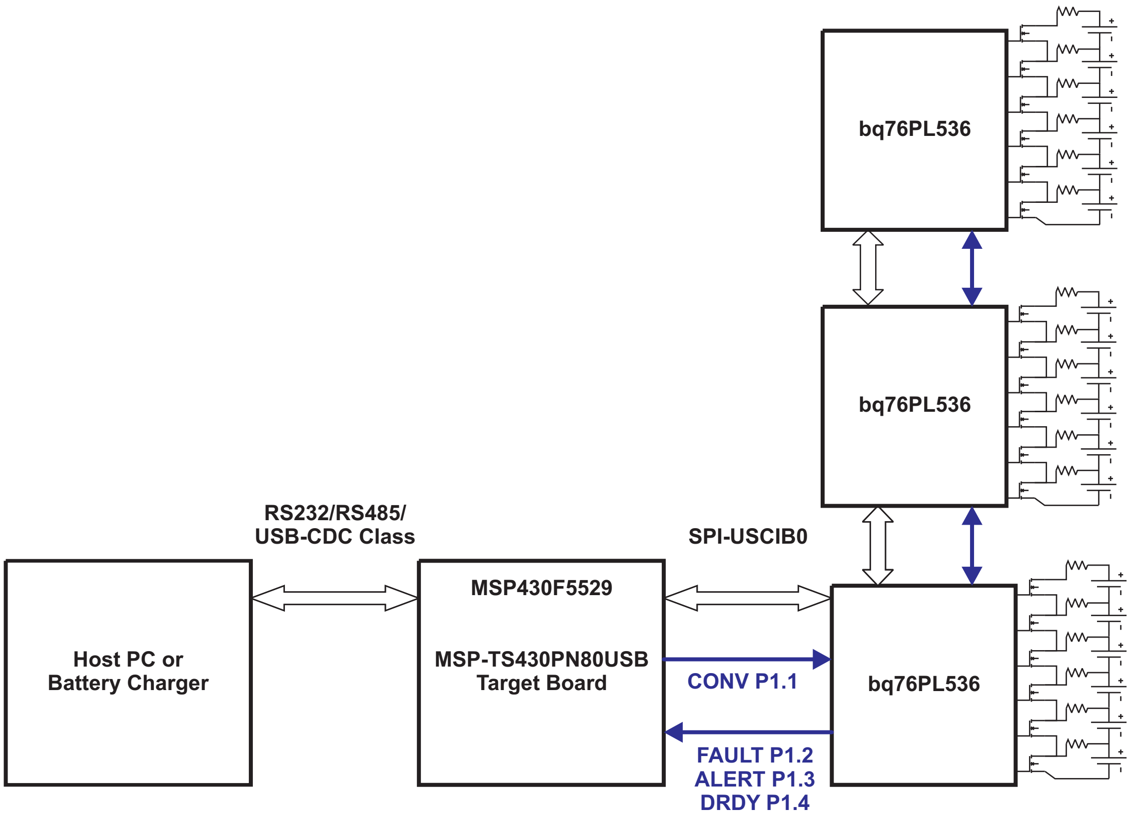

Figure 1. System Block Diagram

1.1 Connecting the Evaluation Modules

The EVMs are connected using standard wire jumpers; Table 1 shows the signal connections between the two EVMs. By default, the MSP430 MCU and isolation circuitry on the BQ76PL536EVM-3 are powered using the USB port on the host computer. Table 2 shows the power connections for powering the boards using the USB VBUS voltage (USB1 connector on the MSP430 EVM). Table 3 shows the jumpers' configuration when the system is powered using the USB VBUS.

The EVMs can also be powered from an external power supply; Table 4 and Table 5 show this power configuration.

NOTE: Make sure that the pullup resistors R49, R53, and R60 are removed on the BQ76PL536EVM-3 board.

Table 1. Connecting the EVMs

Connection Name	BQ76PL536EVM-3 Connector Pin Number : Pin Name	MSP-TS430PN80USB Connector Pin Number : Pin Name
SPI: Select Slave	P5-9 : SPI SS/GPIO	J2-36 : P2.7/USCB0STE
SPI: Slave Input/Master Output	P5-8 : SPI MOSI/GPIO	J2-37 : P3.0/USCB0SIMO
SPI: Slave Output/Master Input	P5-5 : MISO/GPIO	J2-38 : P3.1/USCB0SOMI
SPI: Clock	P5-7 : SCLK/GPIO	J2-39 : P3.2/USCB0CLK
Fault	P4-2 : FAULT	J2-22 : P1.1
Alert	P4-3 : ALERT	J2-23 : P1.2
Data Ready (DRDY)	P4-4 : DRDY	J2-24 : P1.3
Conversion (CONV)	P4-5 : CONV	J2-25 : P1.4

Table 2. EVM Power Connections – USB Powered (USB1 Connector on MSP430™ EVM)

Connection Name	BQ76PL536EVM-3 Connector Pin Number : Pin Name	MSP-TS430PN80USB Connector Pin Number : Pin Name
Power : 5 V from USB bus	P5-6 : SPI+5V	J4-65 : VBUS
Power : Ground	P5-2 : GND2	J5-3 : GND

Table 3. EVMs Jumper Configuration – USB Powered

BQ76PL536EVM-3 Jumper : Pin Selection	MSP-TS430PN80USB Jumper : Pin Selection
JP1 : INT USB	JP1 : 1-2
JP5 : 3.3VDC	JP3 : 2-3
	JP4 : 1-2
	JP2 : 1-2
	JP5 through JP10 : 2-3

Table 4. EVMs Power Connections – 5 V on Power Jack Connector J5 (BQ76PL536)

Connection Name	BQ76PL536EVM-3 Connector Pin Number : Pin Name	MSP-TS430PN80USB Connector Pin Number : Pin Name
Power : 3.3 V	JP5 : PIN 2	J5-1 : VCC
Power : Ground	P5-2 : GND2	J5-3 : GND

Table 5. EVM Jumper Configuration – External Power 5 V on Power Jack Connector J5 (BQ76PL536)

BQ76PL536EVM-3 Jumper : Pin Selection	MSP-TS430PN80USB Jumper : Pin Selection
JP1 : EXT PWR	JP1 : 1-2
JP5 : 3.3VDC	JP3 : 2-3
	JP4 : OPEN
	JP2 : 1-2
	JP5-10 : 2-3

1.2 Communication Ports

The battery management system described in this application report is able to communicate to a host device over USB or asynchronous serial port (UART). To use the USB communication, it is required to connect to the host device using a standard male 4-pin USB-A to USB-B cable between the slave and the master attached to the connector USB1 on the MSP-TS430PN80USB EVM.

It is required to use an external dongle to provide the physical layer for the asynchronous serial port (UART). The user must build or provide an interface for such purpose. A simple RS232 or RS485 interface can be used. The RX and TX should be connected as shown in Table 6.

Table 6. Asynchronous Serial Port (UART) Connections

Connection Name	MSP-TS430PN80USB Connector Pin Number : Pin Name	External Physical Layer
Transmit	J2-40 : P3.3/USCA0TXD	TX
Receive	J3-41 : P3.4/USCA0RXD	RX

2 Software

The battery management system described in this application report is created by configuring, monitoring, and controlling multiple daisy-chained BQ76PL536 devices. These daisy-chained devices create a battery pack that could be built up to 192 Li-Ion cells. The demonstration software created for the application report has the following features:

- Configuring the a multi-stacked battery pack
- Monitoring the multi-cell battery pack voltage and temperature
- Monitoring the individual cell voltage
- Passive cell balancing
- Cell overvoltage and undervoltage protection
- Overtemperature protection
- Charge and discharge mode detection
- Communication to a host device using USB or UART

2.1 Battery Manager

The first task of the battery management software is to initialize the MSP430 MCU peripherals.[5] Then it builds the battery stack by detecting and configuring the existing BQ76PL536. The next tasks are to identify the status of the cells and the battery pack by reading the voltages, temperatures, fault, and alert conditions.

The battery management software is continuously checking for a fail conditions on the battery pack; it samples the cell voltages and the integrity of the battery pack every second. The system goes to low-power mode if there are not any corrective actions or pending tasks. A brief description of this process is shown in Figure 2.

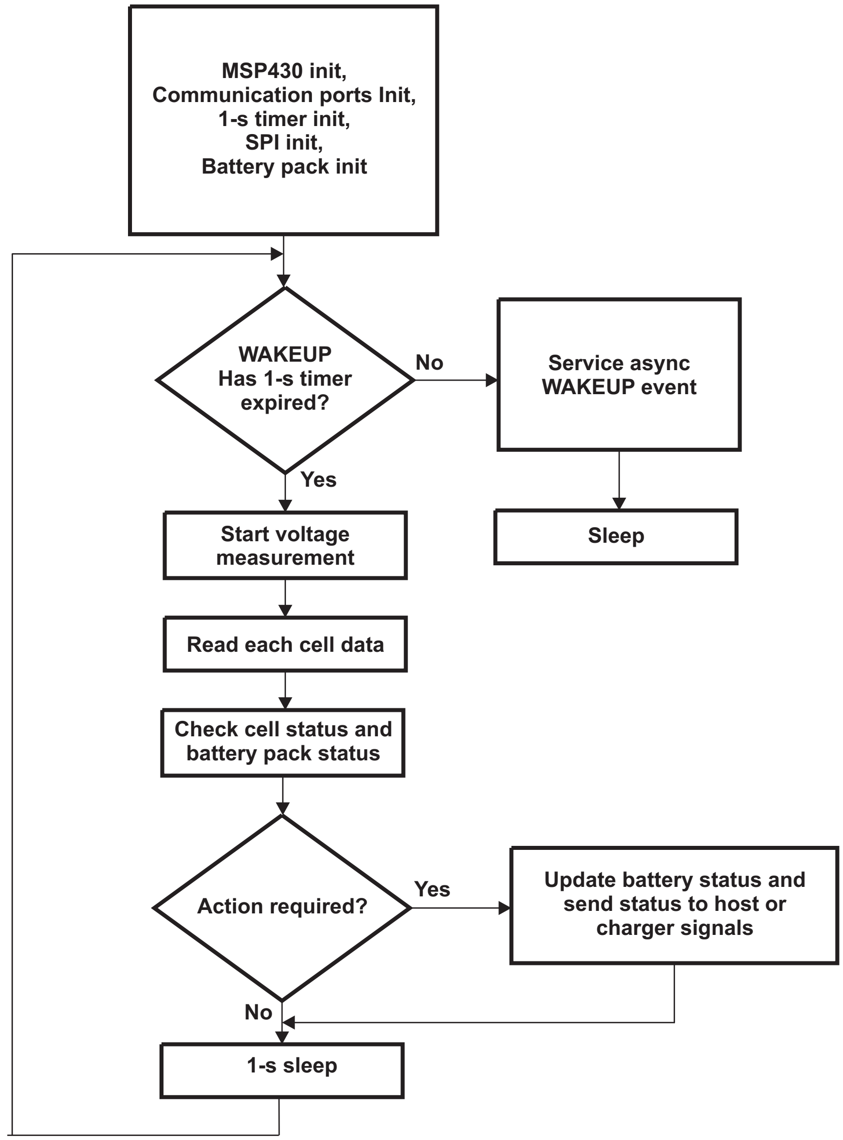

Figure 2. Battery Manager Flowchart

The battery manager software goes into different states depending on the status of the battery pack. Figure 3 shows all of the possible states.

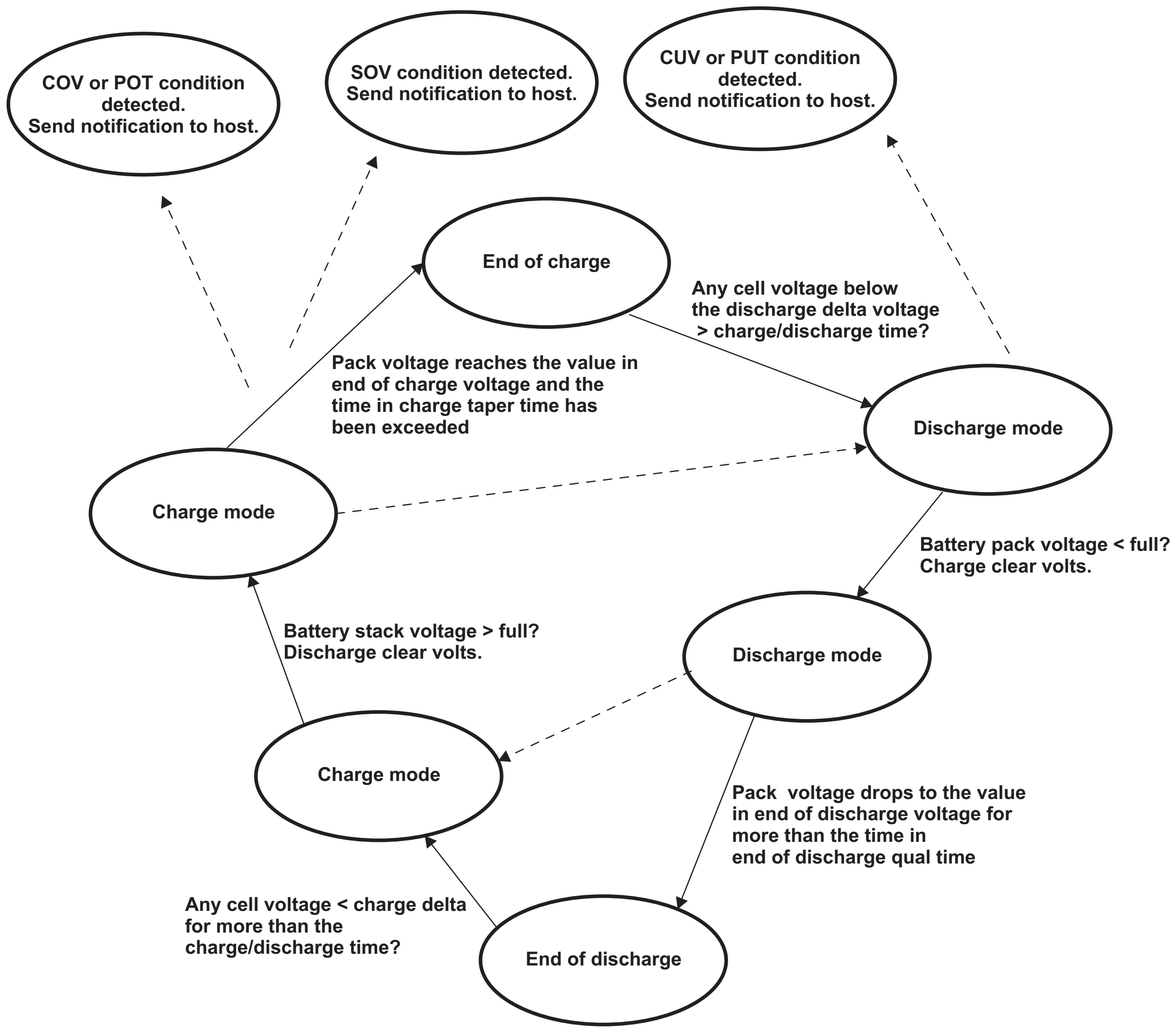

Figure 3. Battery Manager State Machine

2.1.1 Protection and Safety Conditions

The battery manager software protects the battery pack and the cells from undesired operating conditions. To do so, it has some predefined values stored in flash memory to be compared at runtime. These threshold values are located in the data_flash.h file. They are defined based on the Li-Ion cell characteristics as shown in Figure 4. The battery pack threshold values are defined accordingly to the cell threshold values; Figure 5 shows the correlation.

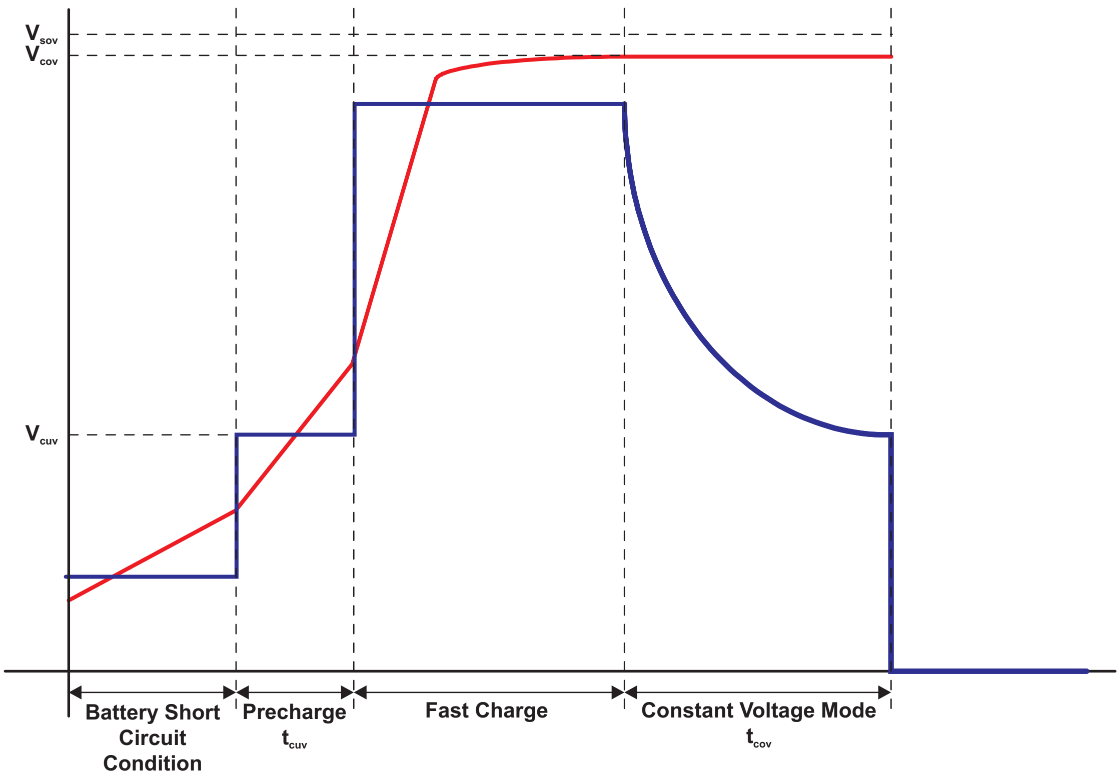

Figure 4. Cell Charging Profile

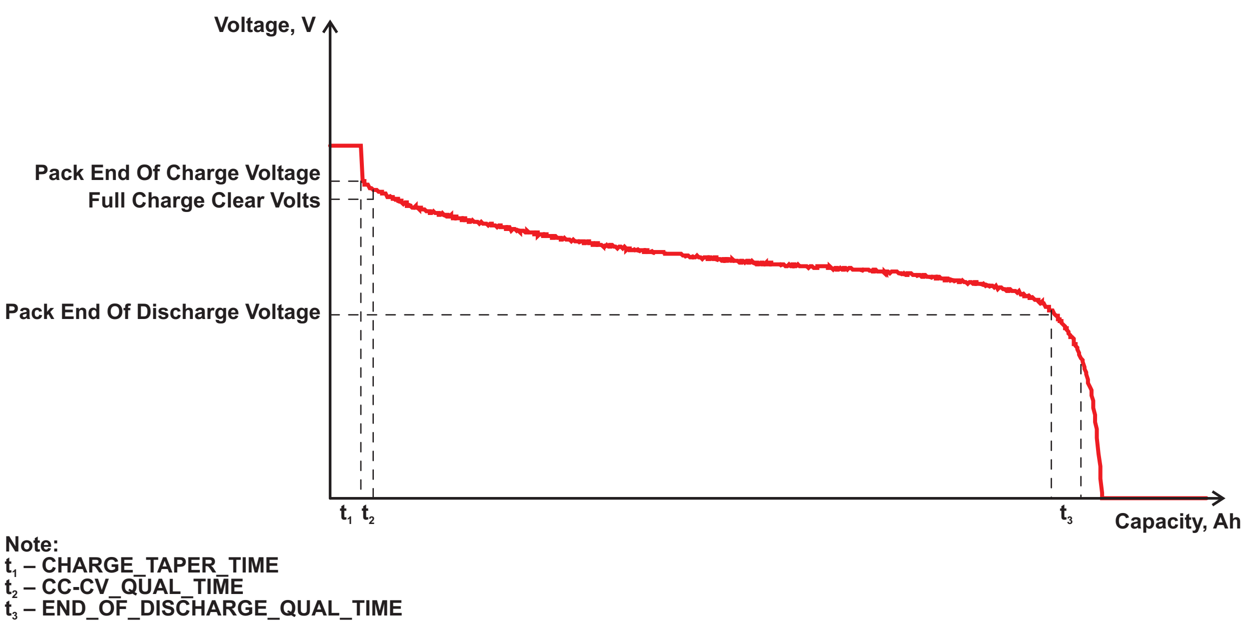

Figure 5. Battery Pack Threshold Values

2.1.2 Cell Balancing

A common problem of multi-cell battery packs is cell imbalance. Cell imbalance is a detriment to the runtime and life span of the whole battery system. To solve this problem, cells with similar state of charge level (SOC) are usually arranged together.[6] However, an imbalance in the state of charge may develop between cells and reduce the overall capacity of the pack. Cell balancing is method that minimizes this problem by equalizing the cells, allowing the battery pack to operate longer.[7]

A difference in cell voltages is the most common manifestation of unbalance, which can be corrected either instantaneously or gradually through bypassing cells with higher voltage. This application report describes a simple method of passive cell balancing based on the differences between the cell voltages. If the difference exceeds the predefined threshold (defined in file data_flash.h), bypass is activated.[8]

This algorithm is active only when the battery pack is in charging mode. It can simultaneously perform multi-cell balancing by determining which cells have to be bypassed based on the voltage difference between the cell with the highest voltage (in the whole pack) and the voltage existing on the cell to be balanced. Figure 6 shows the cell balancing flowchart.

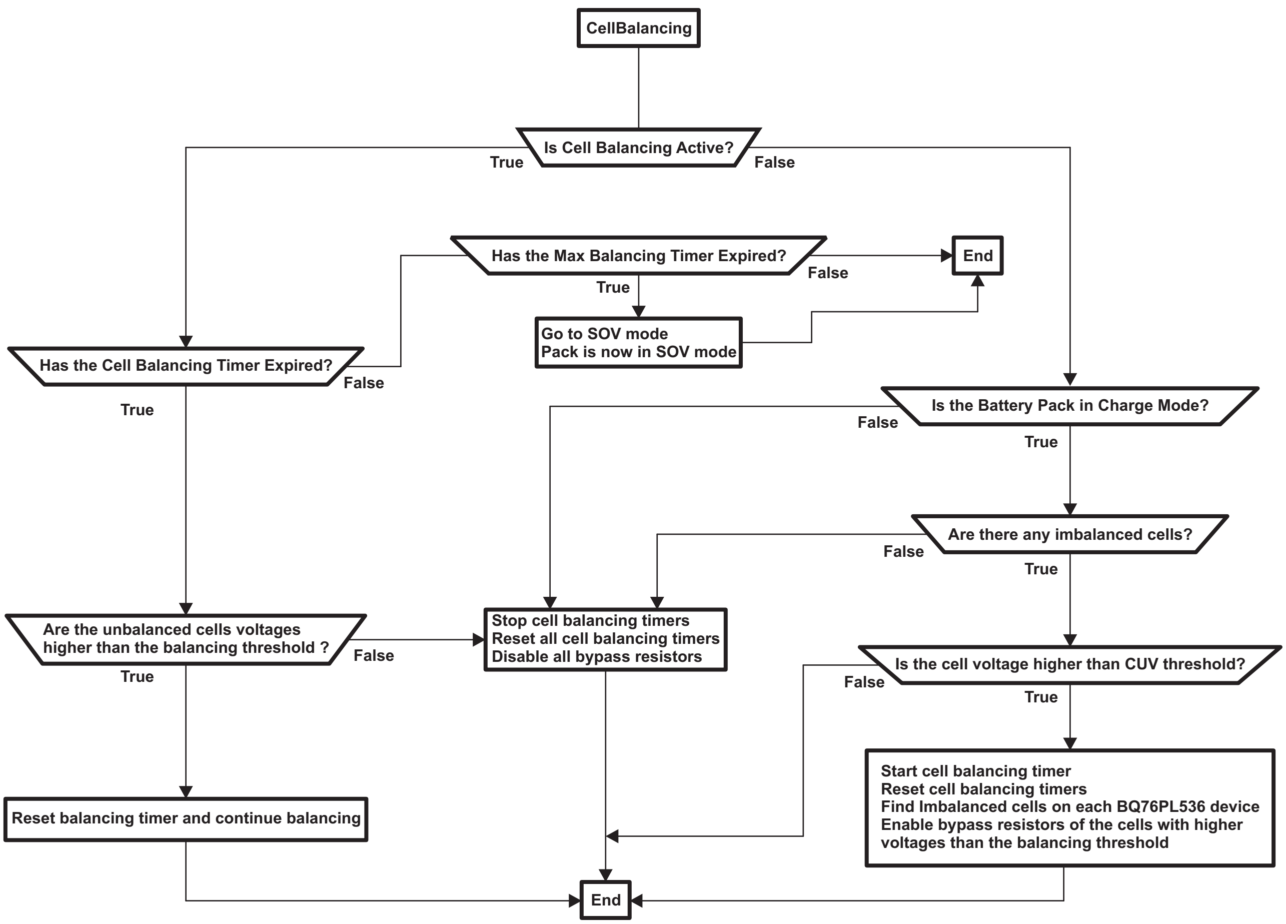

Figure 6. Cell Balancing Flowchart

2.1.3 Communication Ports

The battery management system is capable of reporting the battery pack status by sending data to a host or charger using USB or asynchronous serial protocol (UART). To select any of these communication methods, the user must define it in the main.h file.

The USB module on the MSP430 MCU is configured to act as a USB-CDC device. For more information on the API functions and the structure of the USB stack, see the *Programmer's Guide: MSP430 USB API Stack for CDC/HID*.[9]

The UART API functions are defined in the UART.h, UART.c, and TI_UART_Regs.h. For more information on the specifics of these functions, see INDEX_OPEN_THIS_FILE.html in the folder located in the folder that contains this application report software …/MSP430_BMS_BQ76PL536/docs.

The valid commands accepted by the communication ports are:

"ch" letters + Car Return key sets the battery management system in charge mode

"di" letters + Car Return Key sets the battery management system in discharge mode

"p" letter + Car Return key shows the battery pack status

"c" letter + Car Return Key shows the specific BQ76PL536 device status and cell voltages, entering this command multiple times cycles through the stacked BQ76PL536 devices.

2.1.3.1 Instructions for Interfacing the USB Device to a Windows PC

For each example, open the IAR/CCE workspace, download the project, and run it. Attach the hardware to a PC. If this is the first time attaching hardware equipped with the USB CDC API, you'll probably be prompted for a driver when you attach it. In Windows, direct it to the INF file MSP430_CDC.inf that is located in the root directory of the API. Note that the driver *.sys files are already in Windows; but an INF file must be supplied in order for the device to associated with it. When the installation is completed, start Hyperterminal or another terminal application. Select the appropriate COM port. (After an installation, this is usually the one with the highest number. If necessary, use the Windows Device Manager to identify which COM port is associated with the MSP430 MCU.) The baud rate and other COM port configuration settings do not matter. When working with terminal applications, it will soon be noticed that they weren't written to be tolerant of the COM port disappearing. Whenever the device program is reset while the port is open, or if the cable is removed, and then a re-connect is attempted, the terminal application must close and re-open the port. However, there's an order in which this must be done:

1. Close the terminal's connection to the port.
2. Connect the USB device to the PC; if the PC volume is set to an audible level, an alert can be heard indicating that a new device has been attached.
3. Re-open the terminal's connection to the port.

In short, the port should not be open while the device enumerates. Once learned, this becomes an easy habit. This is merely a symptom of using an old software mechanism (COM ports) over a new medium (USB); terminal applications assume the device is always connected.

NOTE: Sometimes when first attaching the FET tool to the target board, the device can't make a connection. Remove the JTAG connector and USB cable to remove power from the board for a few seconds, then re-connect. Usually it is able to make a connection after this procedure.

2.2 File Structure

Table 7 shows a list describing the contents of each file used in this application report software. For more information on the specifics of these files see INDEX_OPEN_THIS_FILE.html in the folder that contains this application report software /MSP430_BMS_BQ76PL536/docs.

Table 7. Battery Management System Software Files

File	Description
bq_pack.c [code]	This file contains all of the functions of the BQ76PL536 devices
bq_pack.h [code]	This file contains all the definitions of the BQ76PL536 devices
data_flash.c [code]	This file contains all the functions needed to access to the battery pack information stored in the MSP430 MCU flash memory
data_flash.h [code]	This file contains all the definitions of the battery pack
defMSP430USB.h [code]	For more information on this file, see *Programmers Guide: MSP430 USB API Stack for CDC/HID* (available in SLAC285)
descriptors.c [code]	For more information on this file, see *Programmers Guide: MSP430 USB API Stack for CDC/HID* (available in SLAC285)
descriptors.h [code]	For more information on this file, see *Programmers Guide: MSP430 USB API Stack for CDC/HID* (available in SLAC285)
dma.c [code]	For more information on this file, see *Programmers Guide: MSP430 USB API Stack for CDC/HID* (available in SLAC285)
hal_macros.h [code]	This file contains definitions for the Hardware Abstraction Layer (HAL) created for the MSP430 MCU
hal_pmm.c [code]	This file contains definitions for the Hardware Abstraction Layer (HAL) created for the MSP430 MCU
hal_pmm.h [code]	This file contains definitions for the Hardware Abstraction Layer (HAL) created for the MSP430 MCU
hal_UCS.c [code]	This file contains definitions for the Hardware Abstraction Layer (HAL) created for the MSP430 MCU
hal_UCS.h [code]	This file contains definitions for the Hardware Abstraction Layer (HAL) created for the MSP430 MCU
main.c [code]	This file contains the state machine functions and the console functions, It also contains the application functions and interrupt service routines
main.h [code]	This file contains all the definitions of the functions declared in main.c. It also contains the global variables used in other modules
spi_if.c [code]	This file contains the SPI functions and the CRC calculation
spi_if.h [code]	This file contains the SPI functions
TI_IO_Regs.h [code]	Generic I/O ports register definition
TI_USCI_SPI_Regs.h [code]	Provides definitions for the SPI Hardware abstraction layer (HAL)
TI_USCI_UART_Regs.h [code]	Generic UART register definition
TLV.c [code]	This file contains definitions for the Hardware Abstraction Layer (HAL) created for the MSP430 MCU
TLV_descriptors.h [code]	This file contains definitions for the Hardware Abstraction Layer (HAL) created for the MSP430 MCU
types.h [code]	This file contains definitions for the Hardware Abstraction Layer (HAL) created for the MSP430 MCU
UART.c [code]	This file contains all the required functions to send and receive data using the UART module
UART.h [code]	This file contains all the required functions to send and receive data using the UART module
usb.c [code]	For more information on this file, see *Programmers Guide: MSP430 USB API Stack for CDC/HID* (available in SLAC285)
usb.h [code]	For more information on this file, see *Programmers Guide: MSP430 USB API Stack for CDC/HID* (available in SLAC285)
usb_eventHandling.c [code]	For more information on this file, see *Programmers Guide: MSP430 USB API Stack for CDC/HID* (available in SLAC285)
UsbCdc.c [code]	For more information on this file, see *Programmers Guide: MSP430 USB API Stack for CDC/HID* (available in SLAC285)

Table 7. Battery Management System Software Files (continued)

File	Description
UsbCdc.h [code]	For more information on this file, see *Programmers Guide: MSP430 USB API Stack for CDC/HID* (available in SLAC285)
USBCDC_constructs.c [code]	For more information on this file, see *Programmers Guide: MSP430 USB API Stack for CDC/HID* (available in SLAC285)
USBCDC_constructs.h [code]	For more information on this file, see *Programmers Guide: MSP430 USB API Stack for CDC/HID* (available in SLAC285)
UsbIsr.c [code]	For more information on this file, see *Programmers Guide: MSP430 USB API Stack for CDC/HID* (available in SLAC285)
UsbIsr.h [code]	For more information on this file, see *Programmers Guide: MSP430 USB API Stack for CDC/HID* (available in SLAC285)

3 References

1. MSP430F552x data sheet (SLAS590)
2. bq76PL536 data sheet (SLUSA08)
3. *BQ76PL536 EVM Quick Start Guide* (SLUU437)
4. *MSP430™ Hardware Tools User's Guide* (SLAU278)
5. *MSP430x5xx/MSP430x6xx Family User's Guide* (SLAU208)
6. "Cell balancing buys extra run time and battery life" By Sihua Wen, Texas Instrument Inc. (SLYT322)
7. "Battery Cell Balancing: What to Balance and How" By Yevgen Barsukov, Texas Instruments Inc. (http://focus.ti.com/download/trng/docs/seminar/Topic%202%20-%20Battery%20Cell%20Balancing%20-%20What%20to%20Balance%20and%20How.pdf)
8. *Cell Balancing With the bq77PL900* (SLUA463)
9. *Programmer's Guide: MSP430™ USB API Stack for CDC/HID* (available in SLAC285)

Appendix A Benefits of the Ultra-Low-Power MSP430-Based Solution

This appendix describes the benefits and power considerations of using the MSP430 microcontrollers in this multi-cell Li-Ion battery management system.

A.1 Power Modes

This application can run in three power modes. Table 8 through Table 10 show the multiple power schemes and the supply current on each scheme:

- Active Mode

  In this mode, the battery management system is reading the status of the cells and sampling the voltages present at the cells. The BQ76PL536 devices and the MSP430F5529 are in active mode during 25 ms when not sending data to the host; when the battery management system is exchanging information with the host, then the system is active during 100 ms.

Table 8. Supply Currents in Active Mode

Device	Supply Current Typical (mA)	Supply Current Maximum (mA)
BQ76PL536	10.5	15
MSP430F5529	10.1	11

- Low-Power Mode

  In this mode, the BQ76PL536 devices are in protect mode, and the MSP430F5529 is in low-power mode 0. Table 9 shows the supply currents.

Table 9. Supply Currents in Low-Power Mode

Device	Supply Current [(1)] (µA)	Supply Current Maximum [(1)] (µA)
BQ76PL536	45	60
MSP430F5529	83[(2)]	92[(2)]

(1) These currents were measured at 25°C. See the device-specific data sheet for more information.

(2) MSP430F5529 is in low-power mode 3 crystal mode. See the device-specific data sheet for more information

- Ultra-Low-Power Mode

  In this mode, the BQ76PL536 devices are in protect mode and the MSP430F5529 is in low-power mode 3. Table 10 shows the supply currents. Note that USB cannot be used in this mode.

Table 10. Supply Currents in Ultra-Low-Power Mode

Device	Supply Current [(1)] (µA)	Supply Current Maximum [(1)] (µA)
BQ76PL536	45	60
MSP430F5529	2.5[(2)]	3.9[(2)]

(1) These currents were measured at 25°C. See the device-specific data sheet for more information.

(2) MSP430F5529 is in low-power mode 3 crystal mode. See the device-specific data sheet for more information

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DLP® Products	www.dlp.com	Communications and Telecom	www.ti.com/communications
DSP	dsp.ti.com	Computers and Peripherals	www.ti.com/computers
Clocks and Timers	www.ti.com/clocks	Consumer Electronics	www.ti.com/consumer-apps
Interface	interface.ti.com	Energy	www.ti.com/energy
Logic	logic.ti.com	Industrial	www.ti.com/industrial
Power Mgmt	power.ti.com	Medical	www.ti.com/medical
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Space, Avionics & Defense	www.ti.com/space-avionics-defense
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video and Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless-apps

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265