

Christian Cruz, Staff Applications Development Engineer, and **Marvin Neil Cabuenas**, Senior Firmware Engineer

Abstract

The Open Compute Project (OCP), a nonprofit organization that exchanges data center product designs and best practices across businesses, recently published its Open Rack Version 3 (ORV3) specification. The most significant change in the specification is that the design is now migrating from a 12 V to a 48 V architecture. This is Part 2 of a five-part series highlighting Analog Devices' reference design for the battery backup unit (BBU). The first part, "Smart Battery Backup for Uninterrupted Energy Part 1: Electrical and Mechanical Design", discusses the BBU's electrical and mechanical design considerations. Part 2 goes into further detail about the main microcontroller's software, which is responsible for guaranteeing the smooth functioning of processes, hence ensuring the BBU's high efficiency and capacity. Hardware and software must collaborate smoothly to deliver system-level solutions that meet the required specifications.

Introduction

It is imperative to maintain good housekeeping practices in the BBU module as it provides numerous benefits that cannot be ignored. The primary advantage of an orderly and organized circuit is the enhancement and increase in the safety it provides, which reduces the risk of modules catching electrical fires and other hazards. Furthermore, good housekeeping practices improve circuit performance and longevity by preventing electrical faults that can interfere with proper functioning. It is essential to regularly monitor and organize circuits to identify and address any issues or malfunctions that may arise. Prioritizing good housekeeping in circuitry is a simple but effective way to ensure optimal safety and performance, and it should never be overlooked. Each BBU module employs a main microcontroller and a battery management system (BMS) microcontroller. For Analog Devices' reference design, the MAX32690, an ultra low power Arm[®] microcontroller, acts as the main micro-controller and is responsible for six important processes (see Figure 1). A MAX32625 microcontroller for the reference design. The BMS microcontroller is responsible for communicating with the ADBMS6948 chip and passing all measured data to the main microcontroller.

► ANALOG ADI Analog Dialogue™

The main microcontroller's six processes are:

- ▶ Housekeeping and communication via I²C protocol with various peripherals.
- Handling the discharge sequence provided by data from the backplane voltage.
- Handling the charging of the batteries between constant current mode and constant voltage mode.
- Transitioning of the charging and discharging state of the BBU module.
- ► Fault handling and response.
- Responding to Modbus commands as a Modbus follower.



Figure 1. A main controller housekeeping operation cycle.¹

Main Microcontroller Process:

Housekeeping and Communication via I²C with Various Peripherals

With numerous auxiliary devices connected to the module circuit, the primary microcontroller operates as an I²C main. The core microcontroller gathers and saves data from the auxiliary devices while acting as the I²C main. To advance various tasks, the primary microcontroller also controls various I²C ancillary devices. The numerous I²C auxiliary devices are the following:

- BMS microcontroller using the MAX32625.
- LTC2971, a two-channel power system manager.
- MAX31760, a precision fan-speed controller.
- 24AA512T/EEPROM, serving as data storage to retrieve and store important data.
- ▶ LTC2991 temperature sensor.

BMS Microcontroller (MAX32625)

The main microcontroller routinely communicates with the BMS microcontroller (MAX32625) to receive updates on the individual battery pack cell voltages, state of charge (SOC), state of health (SOH), cell temperature, and any faults that may have occurred on the battery pack. Updates are done once every four minutes since it is expected that the cell voltages, SOC, SOH, and temperature will not change rapidly. In case of any fault, a shared pin between the two microcontrollers will flag high and trigger an interrupt on the main microcontroller to immediately read the BMS microcontroller for information on the fault. In the main microcontroller, a dedicated I²C port has been provided to communicate exclusively with the BMS microcontroller to facilitate fast communication between the two microcontrollers.

LTC2971 (Power Management IC)

The main microcontroller routinely communicates with the LTC2971 via the power management bus (PMBUS) protocol checking for voltage, current, and temperature measurements, warnings, and faults. The LTC2971 acts as a quick feedback feature when measuring the backplane output voltage parameters for the main microcontroller to adjust its routine. Also, the device ensures the output voltage is within regulation during the discharge mode of operation by adjusting the feedback voltage of the power converter and allowing it to drop the output voltage by 1%.

MAX31760 (Fan Controller)

The MAX31760 is in charge of regulating the fan speed of the BBU module. The duty cycle of the pulse width modulation is configured by the microcontroller to regulate the speed of the fans through the I²C. The main microcontroller calculates and adjusts the required fan speed based on the temperature and either the backplane load current or the battery pack load current.

24AA512TT (EEPROM/Data Storage)

The on-board EEPROM functions as an external storage device for the entire BBU module. Vital information such as battery voltage levels, SOC, SOH, cell type and model year, and board temperatures are regularly saved to the EEPROM by the main microcontroller through I²C writes to its flash pages. This data is updated every hour and is accessible to users during maintenance and troubleshooting.

LTC2991 (Digital On-Board Temperature Monitoring)

The LTC2991 is an octal voltage, current, and temperature sensor. This device is used to monitor the battery module's temperature from various digital sensors strategically placed inside the module. Based on its temperature readings, the main microcontroller can adjust the fan speed to make sure the working temperature of the power board and the battery stack is maintained and never reaches 40°C.



Figure 2. Constant current and constant voltage charging algorithm.

Handling the Charging of the Batteries Between Constant Current Mode and Constant Voltage Mode

The main microcontroller is also responsible for handling the charging mode of the batteries. At the start, when battery cells are extremely low in voltage capacity, the microcontroller allows a maximum current of 5 A to charge the battery cells. Once the cells are in a steady state condition, the microcontroller will adjust the charging current to be set at 2 A while the voltage per cell has not reached 4 V yet. With constant monitoring of battery cell voltages via data coming from the BMS microcontroller and using an analog-to-digital converter peripheral to measure the overall battery plane voltage, the main microcontroller determines that if all the cells have reached 4 V, the main microcontroller will switch the charging mode to constant voltage mode. This is done by limiting the charging current to only 0.5 A. The main microcontroller will continue monitoring the overall battery plane voltage level while getting data from the BMS microcontroller to check when all batteries are fully charged. This is shown in Figure 2.

Transitioning of the Charge/Discharge State of the Power Converter

It is vital to have a good understanding of how the BBU transitions from standby to battery mode power during a power interruption. To prevent any unexpected power loss and loss of data, the main microcontroller of the module closely monitors the backplane voltage levels. The default charging mode of the LT8228 is set at around 49 V to 53 V, but if the microcontroller detects that the backplane

voltage drops below 48.5 V in 2 ms, it will immediately switch the LT8228 direction pin from charge to discharge mode for four minutes to handle the power outage. The module will continue to discharge until the four minutes are up to ensure uninterrupted power. If the battery cells allow it and the backplane voltage supply has still not returned, the module will wait for one minute to allow the cells to cool down before going into discharge mode. Once power is restored, the unit will switch back to the main power source and start recharging the battery. For specific instructions on how to operate and maintain your BBU, please refer to Figure 3 to understand this process.

Fault Handling and Response

It is essential to have a well-defined plan for handling faults. To avoid stress and frustration, it is crucial to identify potential issues in advance and establish a protocol for addressing them. Clear communication with devices integrated into the system involved is also necessary to ensure that system operation is running smoothly and is aware of what to expect. Remember that mistakes and faults can occur, but how they are dealt with can have a significant impact on the outcome. Thus, the primary microcontroller is specially crafted to reduce the occurrence of false alarms for OCP-specified faults. The firmware program in the microcontroller performs an algorithmic check to detect glitches that trigger faults. After detecting the glitch, it will perform a verification check and will only acknowledge a fault if the problems happen consecutively or within a set number of cycles. This guarantees that only authentic faults are identified and resolved, which is a smart method.



Figure 3. Charging/discharging module transition handling operation.

Responding to a Modbus Command as a Modbus Follower

Modbus function codes are crucial for establishing communication between devices in a Modbus network. They determine the type of request or response sent between devices, such as reading or writing data. Some of the most common Modbus function codes include module metrics, module operation status, module register addresses, and fault status.

The handling of Modbus commands is solely under the control of the main microcontroller. The main microcontroller supports all essential Modbus commands deemed necessary by the OCP specifications. These commands hold the power to alter the module's settings or furnish vital information regarding the battery's SOC, health, cell voltage levels, and charging and discharging currents, among other parameters. Once the main microcontroller verifies the message, it responds in accordance with the command it has received.



Figure 4. A module-to-shelf communication and a shelf-to-PC communication block diagram.

The main microcontroller processes and transmits data from various peripherals to the shelf microcontroller through the UART protocol by utilizing the ADM2561 and ADM3061 transceiver devices, as shown in Figure 4. The advantage of employing

an isolated transceiver is that it is immune to system-level electromagnetic interference and adheres to the OCP-established electromagnetic-compatibility (EMC) standard. Furthermore, the shelf-to-PC communication was accomplished with the ADM2561, which was connected to the host PC via a DB9 (D-Subminiature) connector and an RJ45 internet port.

Summary

An indispensable requirement is a controller architecture that is resolute, productive, and remarkably adaptable. It should have well-defined inputs and outputs, as well as the capacity to handle sophisticated algorithm routines. The architecture should be modular in order to allow for seamless algorithm changes and updates. Furthermore, it should include built-in safety mechanisms to prevent system failures or accidents and easily handle unexpected events. Overall, a high quality controller architecture should provide dependable and effective system control while limiting the possibility of mistakes or malfunctions.

Part 3 of this series will explain the BBU BMS and its associated microcontroller. The article will discuss how readable data is processed and used to maintain the battery pack cell-to-cell voltage and battery pack temperature. Aside from cell balancing, a discussion will be offered on the implementation of the SOC and SOH algorithms and how this information is used to extend the battery pack life cycle.

References

¹David Sun. "Open Rack/SpecsAndDesigns." Open Compute Project.



About the Author

Christian Cruz is a staff applications development engineer at Analog Devices, Inc., Philippines. He holds a bachelor's degree in electronics engineering from the University of the East in Manila, Philippines. He has more than 12 years of engineering experience in the field of analog and digital design, firmware design, and power electronics, which includes power management IC development as well as AC-to-DC and DC-to-DC power conversion. He joined ADI in 2020 and is currently supporting power management requirements for cloud-based computing and system communications applications.



About the Author

Marvin Neil Solis Cabueñas graduated with a bachelor's degree in electronics engineering from De La Salle University in Manila, Philippines. Before joining Analog Devices in 2021, Marvin worked as a systems engineer for Azeus Systems Philippines, Inc., then worked as a network engineer for Technistock, Philippines, Inc. from 2014 to 2017, and as a research and development engineer for Nokia Technology Center Philippines from 2017 to 2020. He has more than nine years of work experience in different fields such as embedded systems programming, digital signal processing, simulation modeling, and others. He currently works as a senior firmware engineer working on various projects for the Power Business Unit of ADI. He is currently working toward his master's degree in electrical engineering at the University of the Philippines.



For regional headquarters, sales, and distributors or to contact customer service and technical support, visit analog.com/contact.

Ask our ADI technology experts tough questions, browse FAQs, or join a conversation at the EngineerZone Online Support Community. Visit ez.analog.com. ©2024 Analog Devices, Inc. All rights reserved. Trademarks and registered trademarks are the property of their respective owners.