User's Guide
BQ79600-Q1 Bring-Up Procedure

TEXAS INSTRUMENTS

ABSTRACT

The BQ79600-Q1 device serves as a bridge device between a host controller and BQ7961x-Q1 battery monitoring devices. With the inclusion of UART and SPI interfaces to communicate to the host, and daisy chain communication ports to talk to BQ7961x-Q1 devices, the BQ79600-Q1 can translate data from the host to the battery monitoring devices and vice versa. This application report details the steps needed to wake up and initialize a stack of BQ7961x-Q1 devices using BQ79600-Q1 as bridge device between the host and the battery monitors.

Table of Contents

List of Figures

Trademarks
All trademarks are the property of their respective owners.

1 Example Wakeup and Initialization Procedure

In this section, a generic *Wakeup and Initialization* procedure is described and can be used as a general reference. As an example, it is assumed that a BQ79600-Q1 is used as a communications bridge, and six BQ79616-Q1 are used to monitor a stack of 96 cells in series. It is assumed that a baud rate of 1 MBaud is used for UART communication to the host, or when using SPI the host provides a clock between 2–6 Mhz. For an example diagram, see Figure 1-1.
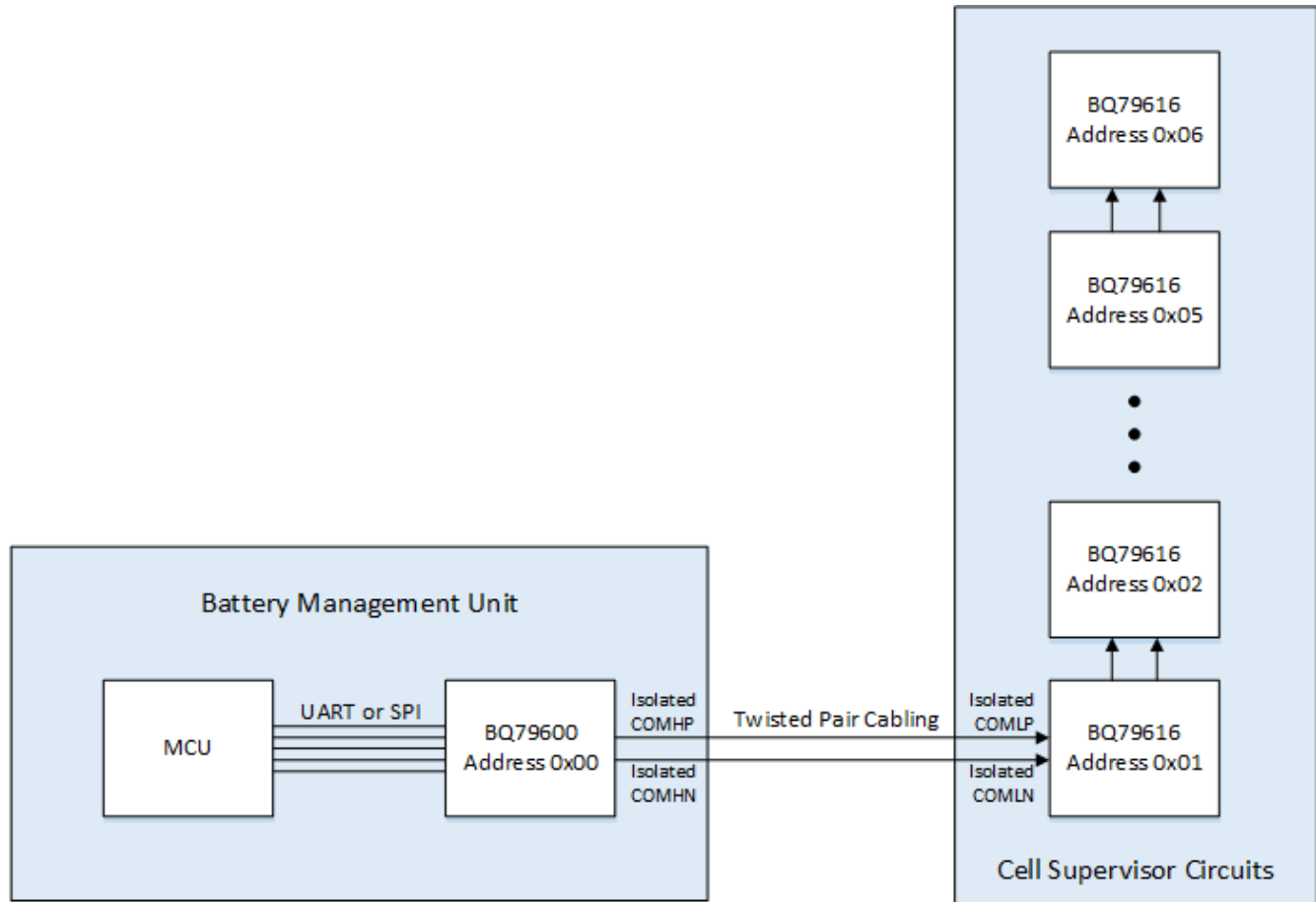

Figure 1-1. Basic System Diagram

There are three required major steps for bringing up a stack of BQ79600-Q1 and BQ79616-Q1 devices:

1. Device Setup
2. Wakeup
3. Auto-Addressing

1.1 Device Setup

In order to set up the BQ79600-Q1 device for communication to the host, it is important to set SPI_RDY and nCS pins in the correct state when using SPI or UART before the device is powered. If UART mode is used, connect SPI_RDY and nCS to GND. If using SPI mode, connect SPI_RDY and nCS to VIO through a 10–100 kΩ pull-up resistor.

To connect the BQ79600-Q1 device to the BQ79616-Q1 device, connect the twisted pair cable as show in Figure 1-1.

1.2 Wakeup Overview

After configuring all required peripherals on the host microcontroller unit (MCU), begin by pulling nCS low so pings can be sent. Furthermore, MOSI or TX signal from the MCU is used to send wake pings to the MOSI/RX pin of the base BQ79600-Q1. Additionally, send commands that instruct the BQ79600-Q1 to propagate a wake tone to the rest of the stacked BQ79616-Q1 devices.

1. Send WAKE ping:
   a. When using SPI, hold nCS low and after 2 us send a WAKE ping on MOSI/RX by pulling the line low for 2.75 ms (t_{HLD_WAKE}) to wake the BQ79600-Q1. Following this, wait 2 us and then bring nCS back high.
   b. When using UART, send a WAKE ping on MOSI/RX by pulling the line low for 2.75 ms (t_{HLD_WAKE}) to wake the BQ79600-Q1.
2. Wait at least 3.5 ms ($t_{SU(WAKE\_SHUT)}$) to allow BQ79600-Q1 device to enter ACTIVE mode.
3. Send a single device write to BQ79600-Q1 to set CONTROL1[SEND_WAKE]=1, which wakes up all stacked BQ79616-Q1 devices.
4. After host completes sending the command, wait appropriate time to allow all devices to receive the WAKE tone and enter ACTIVE mode before starting Auto-Address sequence:
   • The duration of a WAKE tone is about 1.6 ms.
   • The time it takes for a BQ79616-Q1 device to enter ACTIVE mode from shutdown is about 10 ms.

To calculate the total time to wait, add the WAKE tone duration plus the time to enter ACTIVE mode (total of 11.6 ms) and multiply the result by the number of stacked BQ79616-Q1 devices.

Note

If device is shut down through SHUTDOWN ping, the COMH RX and COML RX are disabled at next wake up. In such case, on step 1, host needs to first send a WAKE ping, wait $t_{SU(WAKE\_SHUT)}$ and then send a second WAKE ping. COMH RX and COML RX will be enabled after the second WAKE ping. Then proceed with steps 2 to 4.
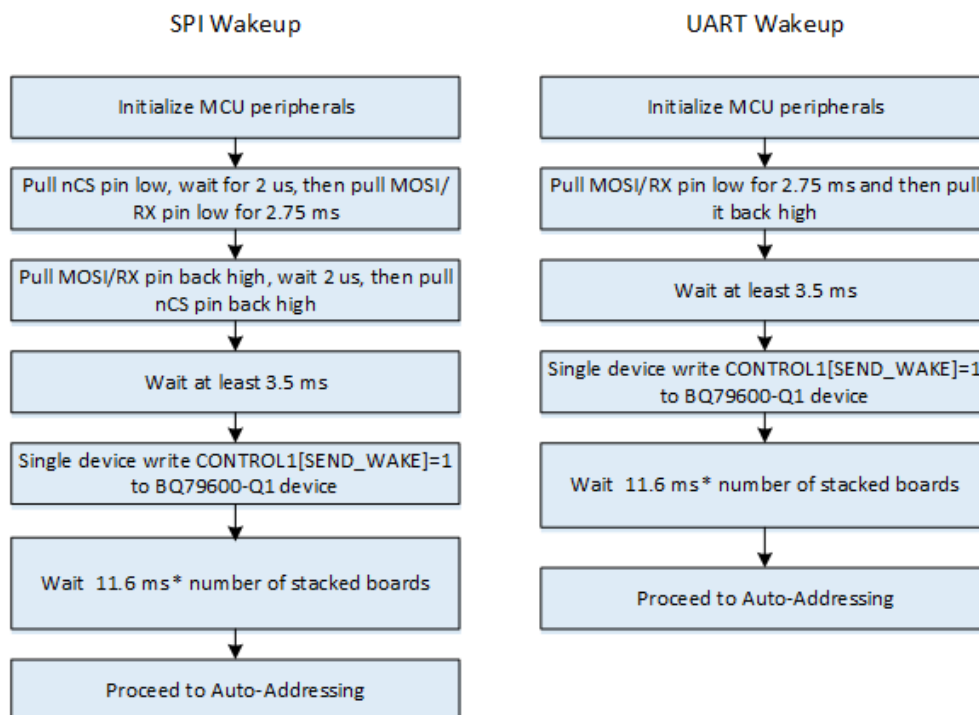

Figure 1-2. Wakeup Flowchart

1.3 Auto-Addressing Overview

Properly addressing the BQ79600-Q1 and each BQ79616-Q1 device in a stack is critical for proper operation of the devices. In addition to the summary of the following main points, all devices must be addressed sequentially up the stack, with a best practice of starting from address 0x00 for the bridge device.

- Follow the WAKEUP procedure to make sure that all devices are awake and available for auto-addressing.
- Dummy stack write 0x00 to registers OTP_ECC_DATAIN1 to OTP_ECC_DATAIN8 (register address 0x343 to 0x34A) on the BQ79616-Q1. This is a total of 8 stack writes (one for each register) to synchronize the DLLs used for reclocking the daisy chain interface.
- Broadcast write 0x01 to register address 0x309 CONTROL1 to enable auto addressing.
- Broadcast write consecutively to register address 0x306 DIR0_ADDR the values 0x00, 0x01, 0x02, and so forth to set the north communication device addresses.
- Broadcast write 0x02 to register address 0x308 to set BQ79616-Q1 as stack devices.
- Single device write 0x03 to register address 0x308 of top device (device 6 in this example) in order to set it as the top of the stack. **Please note that this step is critical for broadcast reads to work properly!**
- Stack read registers OTP_ECC_DATAIN1 to OTP_ECC_DATAIN8 (register address 0x343 to 0x34A). This is a total of 8 stack reads (one for each register) intended to sync up internal DLL.
- Stack read register address 0x306 to verify that all stack devices have been addressed correctly.
- Single device read from the base BQ79600-Q1 device the register address 0x2001 and verify that the value is 0x14.
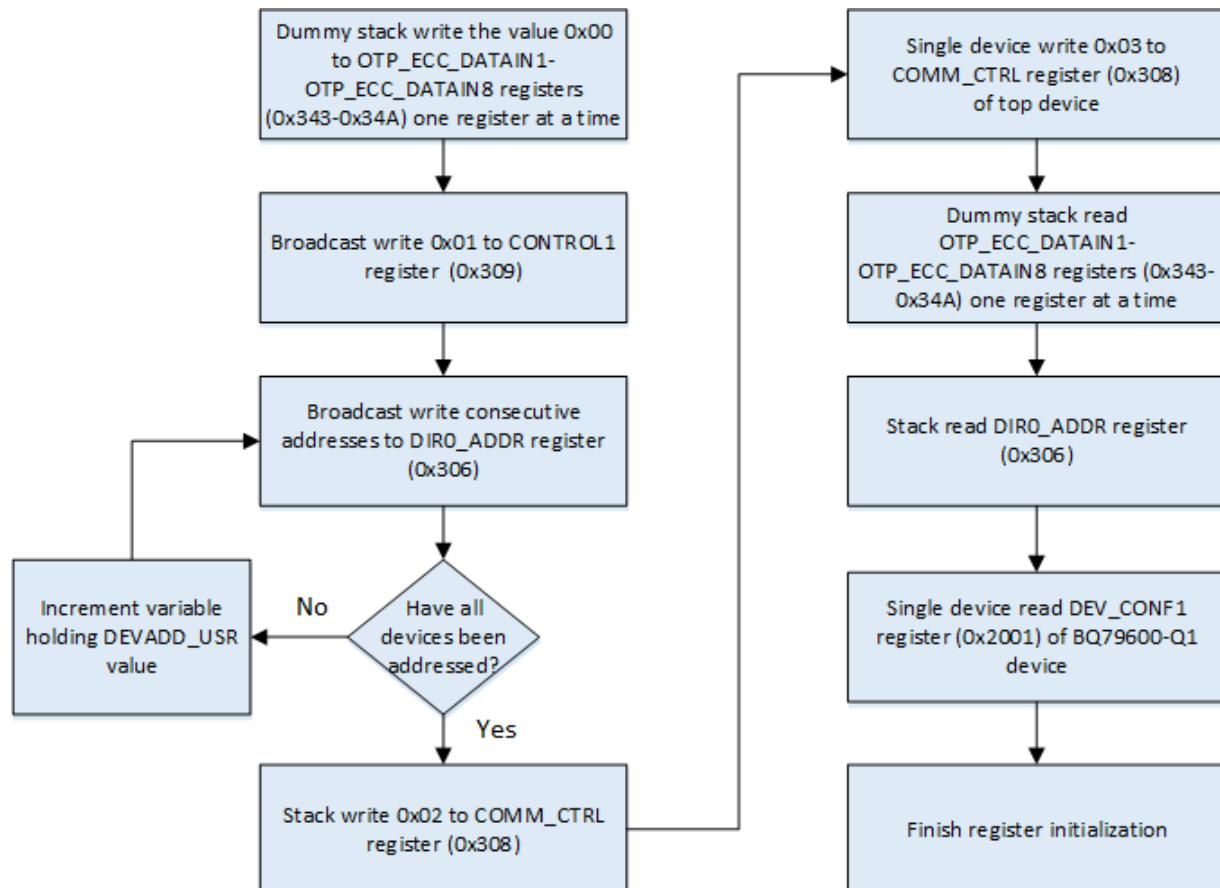- Proceed to register initialization.

Figure 1-3. Auto-Addressing Flowchart

2 References

1. Texas Instruments, *BQ79600-Q1 Automotive SPI/UART Communication Interface Functional-Safety Compliant With Automatic Host Wakeup* data sheet
2. Texas Instruments, *BQ7961x-Q1 Family of 12S, 14S, 16S Precision Automotive Battery Monitor, Balancer and Integrated Hardware Protector* data sheet

IMPORTANT NOTICE AND DISCLAIMER