Application Report

SLUA467B – May 2008 – Revised May 2014

Using I²C Communication with the bq275xx Series of Fuel Gauges

Jeff Mueller and Ming Yu

HVAL - Battery Management Solutions

ABSTRACT

This application report provides two examples that illustrate the bit-transaction details of I²C commands used with the Texas Instruments bq275xx series of battery fuel gauges. The first example shows how to implement a simple fuel gauge command that interrogates the gauge for reading cell voltage. The second example shows the added bit transitions required to access a gauge subcommand, specifically requesting firmware version information. Other I²C commands can be executed in the same manner, using the methodology of these two examples.

Example 1: Reading Cell Voltage

I²C commands are always initiated by the host with a START (S) bit sequence, immediately followed by a 7-bit I²C address with the most-significant bit (MSB) sent first. An eighth bit of 0 is sent by the host, indicating that the next byte to be sent will be a write to the gauge. For the bq275xx series of parts, these 8 bits form the byte 0xAA. Once the start bit and address byte have been successfully received by the gauge, the gauge responds with an ACKNOWLEDGE (A) bit sequence. The gauge is now ready for the subsequent command directive from the host. Further descriptions of the control bit sequences are presented in the Glossary: Control-Bit-Sequence Definitions.

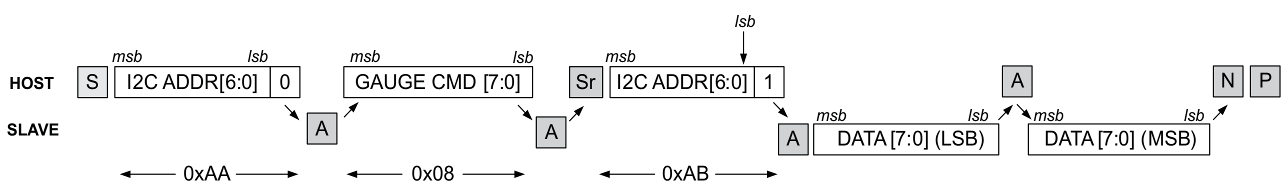

Figure 1. Simple I²C Command Transaction Sequence

After the successful transmission of the I²C address and read/write bit, the gauge command code can be sent by the host, in this case 0x08 for the *Voltage( )* command. The command code is actually a base address location within the gauge and must not be confused with the I²C address for the gauge. Once this location is sent by the host, the gauge responds by sending an ACKNOWLEDGE bit sequence and then executing the corresponding command subroutine.

Even though two-byte locations are used for many gauge commands, writing to only the single byte is required to start gauge command processing. In this case, only 0x08 was written to the gauge, even though the command consists of the two consecutive command bytes 0x08 and 0x09. Most commands consist of two bytes, because the data is returned to these command locations and are also two bytes – the least-significant byte (LSB) is stored in the lower address (0x08), whereas the most-significant byte (MSB) is stored in the higher address (0x09). Like the I²C address data, the gauge command is sent MSB-first.

The host initiates the reading of the command data by sending a REPEAT START (Sr) bit sequence. This is immediately followed by the 7-bit I^{2}C address of the gauge plus the read-bit directive (1), which together create the byte 0xAB. The gauge responds with an ACKNOWLEDGE bit sequence, then takes control of the data bus. The first data byte (LSB) is stored at the 0x08 location in the gauge and is strobed out by the gauge MSB-first. If the host responds with an ACKNOWLEDGE bit sequence, the gauge automatically increments the command location to 0x09, then strobes out the MSB stored there. If the host responds with a NO ACKNOWLEDGE (P or NACK) bit transmission, no further data is spooled to the host. The host terminates the present command packet by sending a STOP bit.

Example 2: Reading the Firmware Version

Reading the firmware version is an example of using the bq275xx subcommands. Subcommands are unique, as they represent another level of depth into the gauge command structure. All subcommands are accessed through the paired command locations at 0x00 and 0x01 in the gauge. The subcommands are written LSB-first. Hence, to send the FW_VERSION subcommand (0x0002), the host writes 0x02 to command location 0x00 and 0x00 to 0x01. Again, I^{2}C always writes the MSB first.

The format for address/command/data exchange between host and gauge is similar to the previous example and is shown in Figure 2. As in Example 1, the host initiates transmission with a START bit, followed by the gauge I^{2}C address and a WRITE bit of 0 (0xAA). The gauge responds with an ACKNOWLEDGE, then the host specifies the command address of 0x00. Again, the gauge responds with an ACKNOWLEDGE.

At this stage, the host must make additional writes to the gauge to set the subcommand code of 0x0002. Hence, the host sends the low byte of the subcommand (0x02). The gauge acknowledges. Then the host sends the high byte of the subcommand (0x00). The gauge issues an ACKNOWLEDGE. The host completes the writing process by issuing the STOP bit sequence. Now the gauge is prepared to return firmware information to the host.

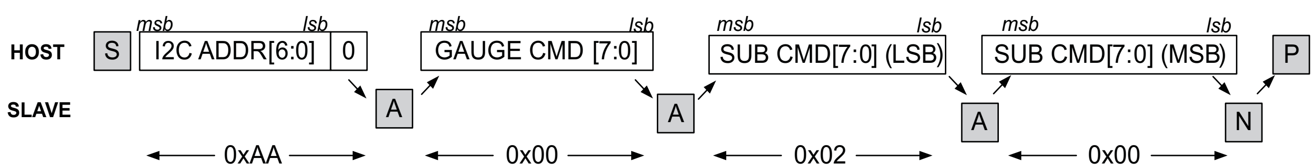

Figure 2. I^{2}C Subcommand Initiation Sequence

To start the reading process, the host proceeds in a manner similar to Example 1, by issuing a START (S) bit sequence, immediately followed by the 7-bit I^{2}C address of the gauge and the eighth bit of 0 (altogether, 0xAA for the bq275xx series of parts). The gauge responds with an ACKNOWLEDGE bit sequence. The host sends the Control( ) command of 0x00, and the gauge acknowledges. The gauge address location has now been set.

To retrieve the data at 0x00 and 0x01, the host proceeds as before. It initiates the reading of the command data by sending a REPEATED START bit sequence. This is immediately followed by the 7-bit I^{2}C address of the gauge plus the read-bit directive (1), which together create the byte 0xAB. The gauge responds with an ACKNOWLEDGE bit sequence, then takes control of the data bus. The first data byte (LSB) is stored at the 0x00 location in the gauge and is strobed out by the gauge MSB-first. If the host responds with an ACKNOWLEDGE bit sequence, the gauge automatically increments the command location to 0x01, then strobes out the MSB stored there. The host terminates the entire command process by sending a STOP bit.

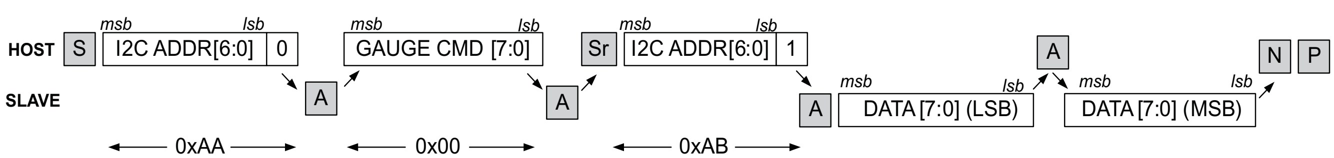

Figure 3. I^{2}C Read Subcommand Sequence

Summary

In summary, remember the following critical aspects when implementing I^{2}C communication between host and bq275xx gauges:

1. Handshaking between the host and gauge is performed by five bit sequences: START, REPEAT START, ACKNOWLEDGE, NO ACKNOWLEDGE, and STOP.
2. The I^{2}C address and read/write bit is the first data item sent at the beginning of an I^{2}C packet transmission. The I^{2}C address must not be mistaken for the command address in the gauge, the latter being the second data field to be transmitted by the host.
3. The host initiates all communication to the gauge and uses the WRITE directive at the end of the I^{2}C address (total byte is 0xAA). When followed by the gauge command, this sets the command address from which the host writes or reads gauge data.
4. When reading or writing multiple bytes, the host should use the base-address auto-increment feature of the gauge, rather than specifying the gauge address location each time a byte is transferred.
5. Whether command address or gauge data, all I^{2}C data is transferred between host and gauge with the least-significant byte first.
6. All bytes transferred between host and gauge is transferred most-significant bit first.

Glossary: Control-Bit-Sequence Definitions

START: The START-bit sequence is a host-generated bit that begins the transmission of every packet. It is defined by a high-to-low transition on the SDA line, while the SCL line is high.

REPEAT START: The REPEAT-START-bit sequence is also a host-generated bit. It has the same characteristics as the START bit, but appears in the middle of a packet transmission. It tells the slave that the gauge command (address) has been specified and data is ready for transfer to/from that command address.

STOP: The STOP-bit sequence indicates the end of a transmission packet. It is consists of a low-to-high transition of the SDA line, while the SCL line is high.

ACKNOWLEDGE: The ACKNOWLEDGE-bit sequence follows every (successful) data field sent between the host and gauge. The device receiving the data field is responsible for sending the bit sequence. The sequence consists of the SDA line being maintained in a low-status, while the SCL line is pulsed high.

NO ACKNOWLEDGE: The NO-ACKNOWLEDGE-bit sequence is frequently used by the host or gauge to indicate the last data byte has been transmitted/received, and that the host should terminate the packet with a STOP-bit sequence. It can also be used to indicate that an I^{2}C device is not listening or capable of responding to the host at a given time. The sequence consists of the SDA line floating in a high state, while the SCL line is pulsed high.

Revision History

Changes from A Revision (May 2009) to B Revision **Page**

- Changed NO ACKNOWLEDGE to ACKNOWLEDGE 2

IMPORTANT NOTICE

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

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