Updating the bq275xx Firmware at Production

Michael Vega and David Maxwell   *PMP - BMS Handheld*

ABSTRACT

The bq275xx devices are gas gauges that are flash based and can be updated, if necessary. Sometimes, however, the gas gauge firmware has been updated, but the system-side gas gauges, such as bq27500 or bq27510, may already be embedded in an enclosed system. This prevents external communication access to the gas gauge device to allow updating with external tools. In this case, it may be useful that the application system main processor perform the actual firmware updating, rather than having an external tool do so. Another situation is a pack-side device implementation, in which the data flash image needs updating, but I2C is no longer accessible and HDQ communication must make the update. This application report provides all the information necessary to implement a system firmware that sends the necessary I2C or HDQ commands to update the firmware or data flash in the bq275xx devices. This application report must be used along with a file of .bqfs type for firmware plus data flash updates or a file of .dffs type for data-flash-only updates. Also available is a tool that allows creating a bqfs and a dffs file with an encrypted record (senc) and optional file containing the specific Data Flash Image (DFI).

Contents

1 Flash Stream File Format

This section describes the file format for the TI BMS FlashStream (.bqfs) and Data Flash FlashStream (.dffs) files. These files contain instructions for I2C or HDQ operations required for updating a device's instruction flash (IF) and/or data flash (DF).

1.1 File Structure

The .bqfs file is an ASCII text file which contains both commands and data. Each line of the file represents one command and potentially 96 bytes of data, as described in the following text. No row contains more than 96 data bytes. The first two characters of each row represent the command, followed by a ":".

"W:" – indicates that the row is a command to write one or more bytes of data.
"R:" – indicates that the row is a command to read one or more bytes of data.
"C:" – indicates that the row is a command to read and compare one or more bytes of data.

"X:" – indicates that the row is a command to wait a given number of milliseconds before proceeding.

White space is used to separate fields within the .bqfs and .dffs files. Each row contains one and only one of the four commands.

The commands discussed in this section can be implemented by a system that can perform multibyte operations for I2C or single-byte operations for I2C and HDQ.

1.2 Write Command

I2C
The write command "W:" instructs the I2C master to write one or more bytes to a given I2C address and given register address. The I2C address format used throughout this document is based on an 8-bit representation of the address. The format of this sequence is:

"W: I2CAddr RegAddr Byte0 Byte1 Byte2…".

For example, the following:

W: AA 55 AB CD EF 00

indicates that the I2C master writes the byte sequence 0xAB 0xCD 0xEF 0x00 to register 0x55 of the device addressed at 0xAA.

More precisely, it indicates to write the following data to the device address 0xAA:

0xAB to register 0x55
0xCD to register 0x56
0xEF to register 0x57
0x00 to register 0x58

HDQ
The write command "W:" instructs the HDQ master to write one byte to a given register address. The format of this sequence is:

"W: RegAddr Byte".

For example, the following:

W: 55 AB

indicates that the HDQ master writes the data 0xAB into register 0x55.

1.3 Read Command

I2C
The read command "R:" instructs the I2C master to read a given number of bytes from a given I2C address and given register address.

The format of this sequence is:

"R: I2CAddr RegAddr NumBytes"

For example, the following:

R: AA 55 100

indicates that the I2C master reads 100 (decimal) bytes of data from register address 0x55 of device address 0xAA.

HDQ

"R: RegAddr NumBytes".

For example, the following:

R: 55 1

indicates that the HDQ master reads 1 byte from register address 0x55.

1.4 Read and Compare Command

I2C

The read and compare command is formatted identically to the write command. The data presented with this command matches the data read exactly, or the operation ceases with an error indication to the user.

The format of this sequence is:

"C: i2cAddr RegAddr Byte0 Byte1 Byte2".

An example of this command is as follows:

C: AA 55 AB CD EF 00

This example expects the master to read back 4 bytes from the register address 0x55 of the device addressed at 0xAA and then compare the data to the values given on the line command in this same order as 0xAB, 0xCD, 0xEF, and 0x00.c.

HDQ

The read and compare consists of making an HDQ read to a given register address and verify that the data received is the same as expected by the FlashStream command.

The format of this sequence is:

"C: RegAddr Byte"".

An example of this command is as follows:

C: 55 AB

This example expects the master to read back the data from register address 0x55 and compare to data 0xAB. An error indication is provided to the user if data is not 0xAB.

1.5 Wait command

The wait command indicates that the host waits a minimum of the given number of milliseconds before continuing to the next row of the flash stream.

For example, the following:

X: 200

indicates that the I2C or HDQ master must wait at least 200 ms before continuing.

2 Firmware Updating Flow

The basic programming flow expected for users to update the firmware and/or data flash of a bq275xx device is summarized in Figure 1. To call on the commands given in the bqfs of dffs file, the user must ensure that the target device is in ROM mode. While in ROM mode, the target device responds to the I2C address of 0x16 (8 bit) or 0x0B (7 bit) if using I2C. From here, the 8-bit I2C address reference is used. To enter ROM mode, 0x0F00 must be written to register address 0x00 of the target device if in the I2C mode, or 0x00 into register 0x00 and 0x0F into register 0x01 if in the HDQ mode. Remember that the I2C address of the device is 0xAA while it is in normal gas gauge mode (default). This step of the process is not part of the bqfs or dffs command sequence. It is up to the user to provide the Enter ROM mode command before using the bqfs or dffs file (Figure 2 and Figure 3).
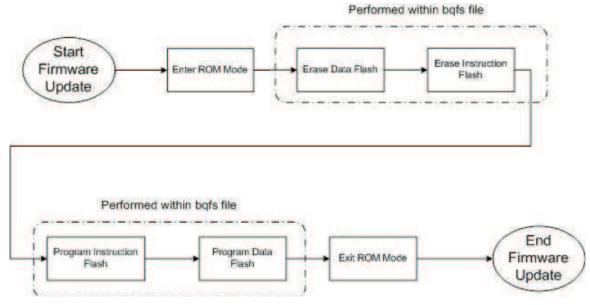
Figure 1. Basic Firmware Programming Flow

Before entering the ROM mode, it must be confirmed that the device is not sealed from a firmware perspective. To verify that the device is not sealed, read Control Status by first writing 0x00 into register 0x00 and 0x01, and then read back register 0x01. If bit 5 is set, then sending the Unseal keys is required. If bit 6 is set, then it requires sending the Full Access Unseal key. These keys are sent by writing the respective keys into the Control register. The keys are 32-bit each. If both keys are required, then they must be entered in the order of Unseal key first and then Full Access Unseal key. For any 32-bit key, no intermittent communication is allowed other than the actual communication to write the keys.
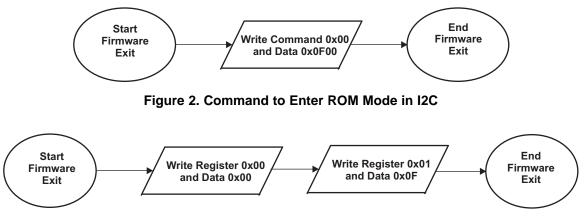

Figure 2. Command to Enter ROM Mode in I2C

Figure 3. Commands to Enter ROM Mode in HDQ

In I2C, it is evident when operating in firmware or ROM mode based on the I2C slave address. If the device sends an Acknowledge signal to the address 0xAA, the device is indicating that it is in firmware mode; if it acknowledges the address 0x16, then it is in ROM mode. Unfortunately, HDQ does not support slave addressing. The following process can indicate in what mode the gas gauge IC is operating:

1. Try to modify register address 0x04. It must be "read only" in firmware mode.

    (a) x = ReadByte(0x04)
    (b) WriteByte(0x04, ~x) // Write back different data
    (c) y = ReadByte(0x04)
    (d) if (x == y) then ~'not in ROM mode'
2. If step 1 indicates that the ROM mode is required, then proceed with the bqfs or dffs sequence.

Once in ROM mode, the user test setup must be able to open the bqfs or dffs file and perform each one of the I2C or HDQ transactions in strictly the same order as given with the FlashStream file. On completing all the commands within the FlashStream file, the user test system sends the Exit ROM mode procedure (Figure 4). At least a 250-ms delay must occur before attempting to proceed with I2C or HDQ communication to the target device using the I2C address 0xAA after exiting ROM Mode.
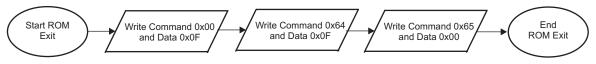

Figure 4. ROM Mode Exit Sequence in I2C or HDQ

3   Debugging bqfs Reader and Programmer

Some development effort is necessary for a user to implement a program that can open a bqfs and give out I2C or HDQ commands based on bqfs format. During the development phase, TI recommends that a user have a dedicated prototype board that allows easy replacement of target devices, This is because during the debugging of the program, it is most likely that devices become useless.

The bqfs sequence is created so as to minimize the chances of hindering any devices in the program debug process and also during the actual production run. During debug, once a target device is placed in the ROM mode, the program developer must not exit the ROM mode unless the full process is completed successfully. Exiting the ROM mode while the firmware or data flash is not fully programmed can place the device in an unrecoverable mode.

The bqfs file contains some sequences defined as Read and Compare. When implementing these commands within the user's program, these commands must be used as checkpoints with which the program developer decides to continue with the process or actually start from the beginning of bqfs sequence. The user must decide how many attempts to make before considering a device unprogrammable.

4   Creating a bqfs and dffs Containing User-Specific DFI

The bqfs file provided with this document is based on the latest firmware revision for a given bq275xx device and its default data flash configuration. A user can have one of two situations in a production environment:

   a) The situation requires programming a specific DFI and retaining the current version of firmware, or

   b) The situation requires programming firmware and also a specific DFI. It is not time-effective for developers to use a bqfs that programs the default firmware and data flash, and then to reprogram the data flash to a custom DFI.

Also, a tool is available that allows converting a senc file into a version that programs custom DFI from the beginning. It is necessary to have a DFI file to use this tool and an original senc file. A senc file is a file that actually is used to reprogram the firmware of a bq275xx device by using the evaluation software. These files are input to the bqfs Update Tool and after executing creates a bqfs and dffs file that contains the desired firmware with the data flash configuration specific to a user's application (Figure 5).

The user has the option to create a bqfs and dffs based on I2C (default) or HDQ. The Update Tool is called from a command screen (DOS) by running the FlashStream.exe file. The command structured for the tool is displayed when calling the FlashStream.exe. The associated files used with the Update Tool must be within the same directory as the tool.

The application report *Going to Production With the bq2750x* (SLUA449) describes how to create a DFI for the first time. The application report *Updating Firmware With the bq2750x and EVM* (SLUA453) describes how to update a DFI that was created with an older version of firmware and now must have a DFI that is compatible with the later version firmware without requiring going through the whole process described in SLUA449.
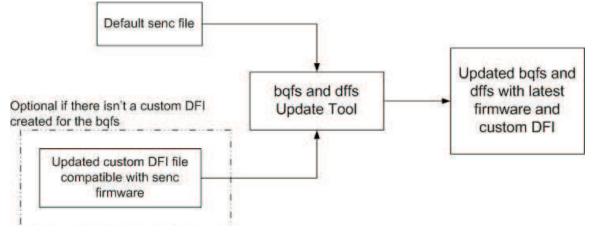
Figure 5. bqfs and dffs Update Tool Process

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	Broadband	www.ti.com/broadband
DSP	dsp.ti.com	Digital Control	www.ti.com/digitalcontrol
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Military	www.ti.com/military
Logic	logic.ti.com	Optical Networking	www.ti.com/opticalnetwork
Power Mgmt	power.ti.com	Security	www.ti.com/security
Microcontrollers	microcontroller.ti.com	Telephony	www.ti.com/telephony
RFID	www.ti-rfid.com	Video & Imaging	www.ti.com/video
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2009, Texas Instruments Incorporated