

bq40zxx Manufacture, Production, and Calibration

Thomas Cosby

ABSTRACT

This application note details manufacture testing, cell voltage calibration, BAT voltage calibration, PACK voltage calibration, current calibration (CC), and temperature calibration for the bq40zxx devices.

Contents

1	Manufacture Testing.....	1
2	Calibration	2
2.1	Cell Voltage Calibration	3
2.2	BAT Voltage Calibration	4
2.3	PACK Voltage Calibration	5
2.4	Current Calibration.....	6
2.5	Temperature Calibration	9

1 Manufacture Testing

To improve the manufacture testing flow, the gas gauge device allows certain features to be toggled on or off through *ManufacturerAccess()* commands. For example, the PRE-CHG FET(), CHG FET(), DS FET(), Lifetime Data Collection(), Calibration() features. Enabling only the feature under test can simplify the test flow in production by avoiding any feature interference. These toggling commands will only set the RAM data, meaning the conditions set by the these commands will be cleared if a reset or seal is issued to the gauge. The *ManufacturingStatus()* keeps track of the status (enabled or disabled) of each feature.

The data flash *ManufacturingStatus* provides the option to enable or disable individual features for normal operation. Upon a reset or a seal command, the *ManufacturingStatus()* will be re-loaded from data flash *ManufacturingStatus()*. This also means if an update is made to *ManufacturingStatus()* to enable or disable a feature, the gauge will only take the new setting if a reset or seal command is sent.

2 Calibration

The device has integrated routines that support calibration of current, voltage, and temperature readings, accessible after writing 0xF081 or 0xF082 to *ManufacturerAccess()* when the *ManufacturingStatus()*[CAL] bit is ON. While the calibration is active, the raw ADC data is available on *ManufacturerData()*. The device stops reporting calibration data on *ManufacturerData()* if any other MAC commands are sent or the device is reset or sealed.

NOTE: The *ManufacturingStatus()*[CAL] bit must be turned OFF after calibration is completed. This bit is cleared at reset or after sealing.

ManufacturerAccess()	Description
0x002D	Enables/Disables <i>ManufacturingStatus()</i> [CAL]
0xF080	Disables raw ADC data output on <i>ManufacturerData()</i>
0xF081	Outputs raw ADC data of voltage, current, and temperature on <i>ManufacturerData()</i>
0xF082	Outputs raw ADC data of voltage, current, and temperature on <i>ManufacturerData()</i> . This mode enables an internal short on the coulomb counter inputs (SRP, SRN).

The *ManufacturerData()* output format is: ZZYYaaAAabbBBccCCddDDeeEEffFFggGGhhHHiillJJkkKK, where:

Value	Format	Description
ZZ	byte	8-bit counter, increments when raw ADC values are refreshed (every 250 ms)
YY	byte	Output status <i>ManufacturerAccess()</i> = 0xF081: 1 <i>ManufacturerAccess()</i> = 0xF082: 2
AAaa	2's comp	Current (coulomb counter)
BBbb	2's comp	Cell voltage 1
CCcc	2's comp	Cell voltage 2
DDdd	2's comp	Cell voltage 3
EEee	2's comp	Cell voltage 4
FFff	2's comp	PACK voltage
Value	Format	Description
GGgg	2's comp	BAT Voltage
HHhh	2's comp	Cell current 1
IIii	2's comp	Cell current 2
JJjj	2's comp	Cell current 3
KKkk	2's comp	Cell current 4

2.1 Cell Voltage Calibration

Figure 1 illustrates cell voltage calibration.

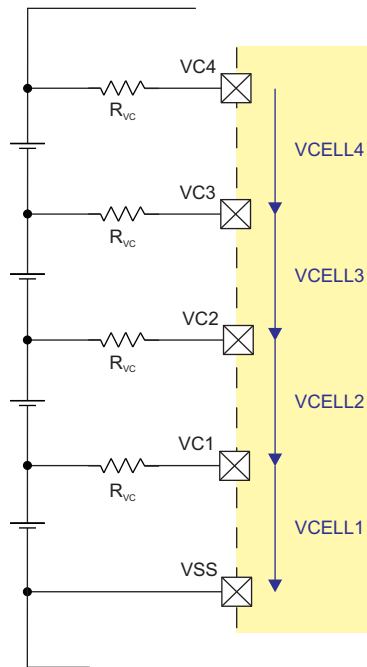


Figure 1. Cell Voltage Calibration

- Apply known voltages in mV to the cell voltage inputs:
 - V_{CELL1} between VC1 pin and VSS pin
 - V_{CELL2} between VC2 pin and VC1 pin
 - V_{CELL3} between VC3 pin and VC2 pin
 - V_{CELL4} between VC4 pin and VC3 pin
- If $ManufacturerStatus()[CAL] = 0$, send 0x002D to $ManufacturerAccess()$ to enable the [CAL] flag.
- Send 0xF081 or 0xF082 to $ManufacturerAccess()$ to enable raw cell voltage output on $ManufacturerData()$.
- Poll $ManufacturerData()$ until the 8-bit counter value increments by 2 before reading data.
- Read the ADC conversion readings of cell voltages from $ManufacturerData()$:

$$ADC_{CELL1} = \text{BBbb of } ManufacturerData()$$
 Is $ADC_{CELL1} < 0x8000$? If yes, use ADC_{CELL1} ; otherwise, $ADC_{CELL1} = -(0xFFFF - \text{BBbb} + 0x0001)$.
- Average several readings for higher accuracy. Poll $ManufacturerData()$ until ZZ increments, to indicate that updated values are available:

$$ADC_{CELL1} = [\text{ADC}_{CELL1}(\text{reading } n) + \dots + \text{ADC}_{CELL1}(\text{reading } 1)]/n$$
- Calculate gain value:

$$\text{Cell Gain} = \frac{V_{CELL1}}{ADC_{CELL1}} \times 2^{16} \tag{1}$$
- Write the new **Cell Gain** value to data flash.
- Re-check voltage readings and if they are not accurate, repeat steps 4 – 6.
- Send 0x002D to $ManufacturerAccess()$ to clear the [CAL] flag if all calibration is complete.

2.2 BAT Voltage Calibration

BAT Voltage Calibration is shown in [Figure 2](#).

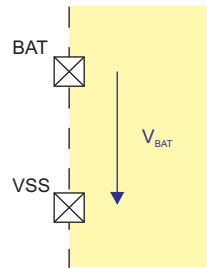


Figure 2. BAT Voltage Calibration

1. Apply known voltages in mV to the voltage input:
 - VBAT between VC4 pin and VSS pin
2. If *ManufacturerStatus()[CAL]* = 0, send 0x002D to *ManufacturerAccess()* to enable the [CAL] flag.
3. Send 0xF081 or 0xF082 to *ManufacturerAccess()* to enable raw cell voltage output on *ManufacturerData()*.
4. Poll *ManufacturerData()* until the 8-bit counter value increments by 2 before reading data.
5. Read ADC conversion readings of cell stack voltage from *ManufacturerData()*:
 - ADCBAT = GGgg of *ManufacturerData()*,
6. Average several readings for higher accuracy. Poll *ManufacturerData()* until ZZ increments to indicate that updated values are available:
 - ADCBAT = [ADCBAT(reading n) + ... + ADCBAT(reading 1)]/n
7. Calculate gain value:

$$\text{BAT Gain} = \frac{V_{\text{BAT}}}{\text{ADCBAT}} \times 2^{16} \quad (2)$$
8. Write the new **BAT Gain** value to data flash.
9. Re-check voltage readings and if they are not accurate, repeat steps 4 – 6.
10. Send 0x002D to *ManufacturerAccess()* to clear the [CAL] flag if all calibration is complete.

2.3 PACK Voltage Calibration

PACK voltage calibration is illustrated in [Figure 3](#).

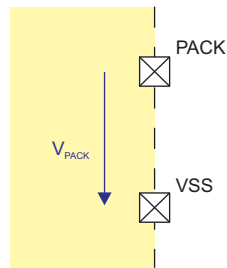


Figure 3. PACK Voltage Calibration

1. Apply known voltages in mV to the voltage input:
 - V_{PACK} between PACK pin and VSS pin
2. If `ManufacturerStatus()[CAL] = 0`, send 0x002D to `ManufacturerAccess()` to enable the [CAL] flag.
3. Send 0xF081 or 0xF082 to `ManufacturerAccess()` to enable raw cell voltage output on `ManufacturerData()`.
4. Poll `ManufacturerData()` until the 8-bit counter value increments by 2 before reading data.
5. Read ADC conversion readings of pack voltage from `ManufacturerData()` :
 - $ADC_{PACK} = \text{FFff}$ of `ManufacturerData()`
6. Average several readings for higher accuracy. Poll `ManufacturerData()` until ZZ increments to indicate that updated values are available:
 - $ADC_{PACK} = [\text{ADC}_{PACK}(\text{reading } n) + \dots + \text{ADC}_{PACK}(\text{reading } 1)]/n$
7. Calculate gain value:

$$\text{PACK Gain} = \frac{V_{PACK}}{ADC_{PACK}} \times 2^{16} \tag{3}$$
8. Write the new **PACK Gain** value to data flash.
9. Re-check voltage readings and if they are not accurate, repeat steps 4 – 6.
10. Send 0x002D to `ManufacturerAccess()` to clear the [CAL] flag if all calibration is complete.

2.4 Current Calibration

A diagram of current calibration is shown in [Figure 4](#).

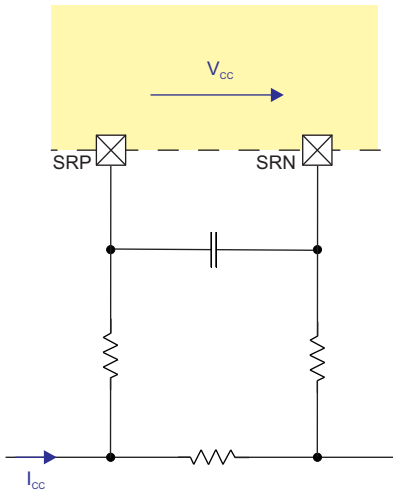


Figure 4. Current Calibration

2.4.1 CC Offset Calibration

NOTE: Due to hardware improvements in this device, CC Offset calibration is not necessary. Only run the CC Offset Calibration procedure if current is observed when no current should be present.

1. Apply a known current of 0 mA, and ensure no current is flowing through the sense resistor connected between the SRP and SRN pins.
2. If *ManufacturerStatus()[CAL]* = 0, send 0x002D to *ManufacturerAccess()* to enable the [CAL] flag.
3. Send 0xF082 to *ManufacturerAccess()* to enable raw cell voltage output on *ManufacturerData()*.
4. Poll *ManufacturerData()* until ZZ increments by 2 before reading data.
5. Obtain the ADC conversion readings of current from *ManufacturerData()*:
 - $ADC_{CC} = \text{AAaa of } ManufacturerData()$
 - Is $ADC_{CC} < 0x8000$? If yes, use ADC_{CC} ; otherwise, $ADC_{CC} = -(0xFFFF - \text{AAaa} + 0x0001)$.
6. Average several readings for higher accuracy. Poll *ManufacturerData()* until ZZ increments to indicate that updated values are available:
 - $ADC_{CC} = [\text{ADC}_{CC}(\text{reading } n) + \dots + \text{ADC}_{CC}(\text{reading } 1)]/n$
7. Read *Coulomb Counter Offset Samples* from data flash.
8. Calculate offset value:
 - $\text{CC offset} = ADC_{CC} \times (\text{Coulomb Counter Offset Samples})$
9. Write the new *CC Offset* value to data flash.
10. Re-check the current reading and if it is not accurate, repeat steps 1 – 10.
11. Send 0x002D to *ManufacturerAccess()* to clear the [CAL] flag if all calibration is complete.

2.4.2 Board Offset Calibration

NOTE: Due to hardware improvements in this device, Board Offset calibration is not necessary. Only run the Board Offset Calibration procedure if board offset current is observed.

1. Ensure that Offset Calibration was performed first.
2. Apply a known current of 0 mA, and ensure no current is flowing through the sense resistor connected between the SRP and SRN pins.
3. If *ManufacturerStatus()[CAL]* = 0, send 0x002D to *ManufacturerAccess()* to enable the [CAL] flag.
4. Send 0xF081 to *ManufacturerAccess()* to enable raw cell voltage output on *ManufacturerData()*.
5. Poll *ManufacturerData()* until ZZ increments by 2 before reading data.
6. Obtain the ADC conversion readings of current from *ManufacturerData()*:
 - $ADC_{CC} = \text{AAaa of } ManufacturerData()$
Is $ADC_{CC} < 0x8000$? If yes, use ADC_{CC} ; otherwise, $ADC_{CC} = -(0xFFFF - \text{AAaa} + 0x0001)$.
7. Average several readings for higher accuracy. Poll *ManufacturerData()* until ZZ increments to indicate that updated values are available:
 - $ADC_{CC} = [\text{ADC}_{CC}(\text{reading } n) + \dots + \text{ADCCC}(\text{reading } 1)]/n$
8. Read *Coulomb Counter Offset Samples* from data flash.
9. Calculate offset value:
 - Board offset = $(ADC_{CC} - \text{CC Offset}) \times \text{Coulomb Counter Offset Samples}$
10. Write the new *Board Offset* value to data flash.
11. Re-check the current reading. If the reading is not accurate, repeat steps 1 – 10.
12. Send 0x002D to *ManufacturerAccess()* to clear the [CAL] flag if all calibration is complete.

2.4.3 CC Gain/Capacity Gain Calibration

1. Apply a known current (typically 1 A to 2 A), and ensure ICC is flowing through the sense resistor connected between the SRP and SRN pins.
2. If *ManufacturerStatus()[CAL]* = 0, send 0x002D to *ManufacturerAccess()* to enable the [CAL] flag.
3. Send 0xF081 to *ManufacturerAccess()* to enable raw CC output on *ManufacturerData()*.
4. Poll *ManufacturerData()* until ZZ increments by 2 before reading data.
5. Read the ADC conversion readings of current from *ManufacturerData()*:
 - $ADC_{CC} = \text{AAaa of } ManufacturerData()$
Is $ADC_{CC} < 0x8000$? If yes, use ADC_{CC} ; otherwise, $ADC_{CC} = -(0xFFFF - \text{AAaa} + 0x0001)$.
6. Average several readings for higher accuracy. Poll *ManufacturerData()* until ZZ increments to indicate that updated values are available:
 - $ADC_{CC} = [ADC_{CC}(\text{reading } n) + \dots + ADC_{CC}(\text{reading } 1)]/n$
7. Read **Coulomb Counter Offset Samples** from data flash.
8. Calculate gain values:

$$CC \text{ Gain} = \frac{I_{CC}}{ADC_{CC} - \frac{\text{Board Offset} + \text{CC Offset}}{\text{Coulomb Counter Offset Samples}}}$$

$$\text{Capacity Gain} = CC \text{ Gain} \times 298261.6178 \quad (4)$$

9. Write the new **CC Gain** and **Capacity Gain** values to data flash.
10. Re-check the current reading. If the reading is not accurate, repeat steps 1 – 9.
11. Send 0x002D to *ManufacturerAccess()* to clear the [CAL] flag if all calibration is complete.

NOTE: There is a conversion factor for CC Gain and Capacity Gain parameters entered in bqStudio.

Name	Data Type	Data Flash Default	bqStudio Default	DF-to-Studio Conversion
CC Gain	F4	3.58422	1.036	3.714528/DF
Capacity Gain	F4	1069035.256	1.036	1107901.13/DF

2.5 Temperature Calibration

Figure 5 illustrates temperature calibration.

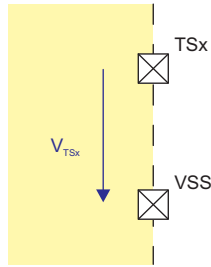


Figure 5. Temperature Calibration

2.5.1 Internal Temperature Sensor Calibration

1. Apply a known temperature in 0.1°C, and ensure that temperature $TEMP_{TINT}$ is applied to the device.
2. Read the TINT $offset_{old}$ from **Internal Temp Offset**.
3. Read the reported temperature from `DAStatus2()`:
 - $TINT = AA_{aa}$ of `DAStatus2()`
 - Is $TINT > 0$? If yes, $TINT = AA_{aa} - 2732$.
4. Calculate temperature offset:

$$TINT\ offset = TEMP_{TINT} - TINT + TINT\ offset_{old} \quad (5)$$
5. Write the new **Internal Temp Offset** value to data flash.
6. Re-check the `DAStatus2()` reading. If the reading is not accurate, repeat steps 1 – 5.

2.5.2 TS1–TS2–TS3–TS4 Calibration

1. Apply a known temperature in 0.1°C, and ensure that temperature $TEMP_{TSx}$ is applied to the thermistor connected to the TSx pin. "TSx" refers to TS1, TS2, TS3, or TS4, whichever is applicable.
2. Read the TSx $offset_{old}$ from **External x Temp Offset**, where x is 1, 2, 3, or 4.
3. Read the appropriate temperature from the `DAStatus2()` block as TSx.
4. Calculate the temperature offset:

$$TSx\ offset = TEMP_{TSx} - TSx + TSx\ offset_{old} \quad (6)$$

Where x is 1, 2, 3, or 4.
5. Write the new **External x Temp Offset** (where x is 1, 2, 3, or 4) value to data flash.
6. Re-check the `DAStatus2()` reading. If the reading is not accurate, repeat steps 1 – 5.

Revision History

Changes from Original (March 2015) to A Revision	Page
• Added '+' sign to value of ADC_{CELL1} in number 5 in the <i>Cell Voltage Calibration</i> section.	3
• Deleted factor ' V_{CELL1} ' from Equation 1.	3
• Added '+ J +' to '...' in step 6 of <i>BAT Voltage Calibration</i> section.....	4
• Changed '+ J +' to '...' in number 6 in the <i>PACK Voltage Calibration</i> section.	5

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as "components") are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

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

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

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license 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 significant portions 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. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

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

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have **not** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products

Audio	www.ti.com/audio
Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DLP® Products	www.dlp.com
DSP	dsp.ti.com
Clocks and Timers	www.ti.com/clocks
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com
RFID	www.ti-rfid.com
OMAP Applications Processors	www.ti.com/omap
Wireless Connectivity	www.ti.com/wirelessconnectivity

Applications

Automotive and Transportation	www.ti.com/automotive
Communications and Telecom	www.ti.com/communications
Computers and Peripherals	www.ti.com/computers
Consumer Electronics	www.ti.com/consumer-apps
Energy and Lighting	www.ti.com/energy
Industrial	www.ti.com/industrial
Medical	www.ti.com/medical
Security	www.ti.com/security
Space, Avionics and Defense	www.ti.com/space-avionics-defense
Video and Imaging	www.ti.com/video

TI E2E Community

e2e.ti.com