bq40z80 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 bq40z80 device.

Contents

Trademarks

All trademarks are the property of their respective owners.

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(), PRE-DSG 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 only set the RAM data, meaning the conditions set by the these commands is 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()* is re-loaded from data flash *ManufacturingStatus()*. This also means if an update is made to *ManufacturingStatus()* to enable or disable a feature, the gauge only takes 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 or 0xF083 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 *ManufacturingStatus()* [CAL]
0xF080	Disables raw ADC data output on *ManufacturerData()*
0xF081	Outputs raw ADC data of voltage, current, and temperature on *ManufacturerData()*
0xF082	Outputs raw ADC data of voltage, current, and temperature on *ManufacturerData()*. This mode enables an internal short on the coulomb counter inputs (SRP, SRN).
0xF083	Outputs raw ADC data of cell-7, voltage and current on *ManufacturerData()*

The *ManufacturerData()* output format for 0xF081 and 0xF082 is: ZZYYaaAAbbBBccCCddDDeeEEffFFggGGhhHHiiIIjjJJkkKKllLLmmMMnnNNooOO,

where:

Value	Format	Description
ZZ	byte	8-bit counter, increments when raw ADC values are refreshed (every 250 ms)
YY	byte	Output status *ManufacturerAccess()* = 0xF081: 1 *ManufacturerAccess()* = 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	Cell voltage 5
GGgg	2's comp	Cell voltage 6
HHhh	2's comp	PACK voltage
IIii	2's comp	BAT Voltage
JJjj	2's comp	Cell current 1
KKkk	2's comp	Cell current 2
LLll	2's comp	Cell current 3
MMmm	2's comp	Cell current 4
NNnn	2's comp	Cell current 5
OOoo	2's comp	Cell current 6

The *ManufacturerData()* output format for 0xF083 is: ZZYYaaAAbbBB, where:

Value	Format	Description
ZZ	byte	8-bit counter, increments when raw ADC values are refreshed (every 250 ms)
YY	byte	Output status *ManufacturerAccess()* = 0xF083: 1
AAaa	2's comp	Cell voltage 7
BBbb	2's comp	Cell current 7

2.1 Cell Voltage 1-6 Calibration

Figure 1 illustrates cell voltage calibration.

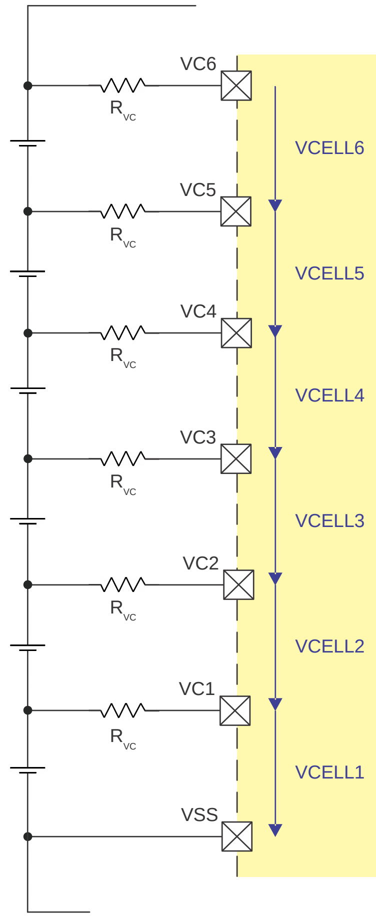

Figure 1. Cell 1-6 Voltage Calibration

1. 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
   - V_{CELL5} between VC5 pin and VC4 pin
   - V_{CELL6} between VC6 pin and VC5 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 the ADC conversion readings of cell voltages from *ManufacturerData()*:
   - ADC_{CELL1} = BBbb of *ManufacturerData()*
   - Is ADC_{CELL1} < 0x8000? If yes, use ADC_{CELL1}; otherwise, ADC_{CELL1} = –(0xFFFF – BBbb + 0x0001).

6. Average several readings for higher accuracy. Poll *ManufacturerData()* until ZZ increments, to indicate that updated values are available:
   - ADC_{CELL1} = [ADC_{CELL1}(reading n) + … + ADC_{CELL1}(reading 1)]/n
7. Calculate gain value:

$$Cell\ \ Gain = \frac{V_{CELL1}}{ADC_{CELL1}} \times 2^{16} \quad (1)$$

8. Write the new ***Cell 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.

2.2 VC7 Sense Gain Calibration

The differential voltage of Cell-7 is determined by subtracting two single-ended measurements:

- The 7P voltage (top of Cell-7) is measured using an external resistor divider from 7P to VSS, with the divided-down voltage applied to Pin-12 (RC2) VC7SENSE
- The 6P voltage (bottom of Cell-7) is measured using the internal BAT measurement of VC6-VSS

In order to obtain an accurate Cell-7 differential voltage measurement, it is necessary to calibrate the gain of the external resistor divider. The measurement for VC7 sense gain is done from the VC7SENSE pin using the external voltage divider.

VC7 Sense Gain calibration is illustrated in Figure 2.

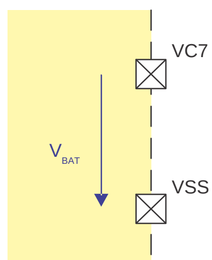

Figure 2. VC7 Sense Gain Calibration

1. Apply known voltages in mV to the voltage input:
   - VC7-VSS between VC7SENSE pin and VSS pin
2. If *ManufacturerStatus()[CAL]* = 0, send 0x002D to *ManufacturerAccess()* to enable the [CAL] flag.
3. Send 0xF083 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()* :
   - $ADC_{VC7\text{-}VSS}$ = AAaa of *ManufacturerData()*
6. Average several readings for higher accuracy. Poll *ManufacturerData()* until ZZ increments to indicate that updated values are available:
   - $ADC_{VC7\text{-}VSS}$ = [$ADC_{VC7\text{-}VSS}$(reading n) + … + $ADC_{VC7\text{-}VSS}$(reading 1)]/n
7. Calculate gain value:

$$V_{C7}Sense\ \ Gain = \frac{VC7 - V_{SS}}{ADC_{VC7} - V_{SS}}\ 2^{16} \quad (2)$$

8. Write the new ***VC7 Sense 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 VC6-VSS Voltage Calibration

VC6-VSS (BAT in case of 7th cell is not used) Voltage Calibration is shown in Figure 3.

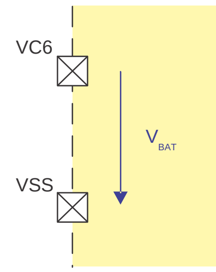

Figure 3. VC6-VSS Gain Calibration

1. Apply known voltages in mV to the voltage input:
   - VC6-VSS between VC6 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()*:
   - $ADC_{VC6\text{-}VSS}$ = iiII of *ManufacturerData()*
6. Average several readings for higher accuracy. Poll *ManufacturerData()* until ZZ increments to indicate that updated values are available:
   - $ADC_{VC6\text{-}VSS}$ = [$ADC_{VC6\text{-}VSS}$(reading n) + … + $ADC_{VC6\text{-}VSS}$(reading 1)]/n
7. Calculate gain value:

$$VC6 - V_{SS}\ Gain = \frac{VC6 - V_{SS}}{ADC_{VC6} - V_{SS}} \times 2^{16} \quad (3)$$

8. Write the new ***VC6 - VSS 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 PACK Voltage Calibration

PACK voltage calibration is illustrated in Figure 4.

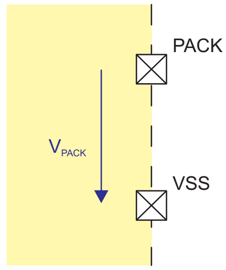

Figure 4. PACK Voltage Calibration

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} = HHhh of *ManufacturerData()*
6. Average several readings for higher accuracy. Poll *ManufacturerData()* until ZZ increments to indicate that updated values are available:
   - $ADC_{PACK} = [ADC_{PACK}(\text{reading } n) + \ldots + ADC_{PACK}(\text{reading } 1)]/n$
7. Calculate gain value:

$$PACK\ \ Gain = \frac{V_{PACK}}{ADC_{PACK}} \times 2^{16} \quad (4)$$

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.

2.5 Current Calibration

A diagram of current calibration is shown in Figure 5.

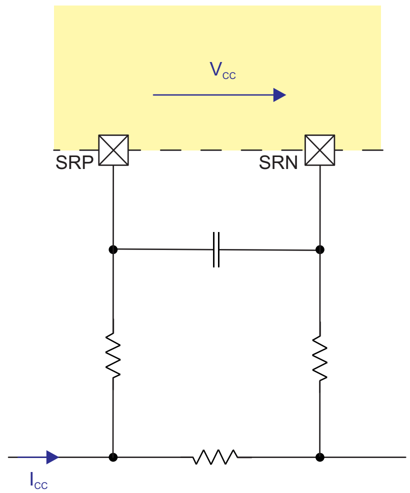

Figure 5. Current Calibration

2.5.1 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} = AAaa of *ManufacturerData()*

   Is ADC_{CC} < 0x8000? If yes, use ADC_{CC}; otherwise, ADC_{CC} = –(0xFFFF – 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}(reading n) + … + ADC_{CC}(reading 1)]/n
7. Read *Coulomb Counter Offset Samples* from data flash.
8. Calculate offset value:
   - CC offset = ADC_{CC} × (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.

2.5.2 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} = AAaa of *ManufacturerData()*

   Is ADC_{CC} < 0x8000? If yes, use ADC_{CC}; otherwise, ADC_{CC} = –(0xFFFF – AAaa + 0x0001).
7. Average several readings for higher accuracy. Poll *ManufacturerData()* until ZZ increments to indicate that updated values are available:
   - ADC_{CC} = [ADC_{CC}(reading n) + … + ADCCC(reading 1)]/n
8. Read *Coulomb Counter Offset Samples* from data flash.
9. Calculate offset value:
   - Board offset = (ADC_{CC} – CC Offset) × 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.

2.5.3 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} = AAaa of *ManufacturerData()*

   Is ADC_{CC} < 0x8000? If yes, use ADC_{CC}; otherwise, ADC_{CC} = –(0xFFFF – 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}(reading n) + … + ADC_{CC}(reading 1)]/n
7. Read ***Coulomb Counter Offset Samples*** from data flash.
8. Calculate gain values:

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

$$Capacity\ Gain = CC\ Gain\ x\ 298261.6178 \qquad (5)$$

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

2.6 Temperature Calibration

Figure 6 illustrates temperature calibration.

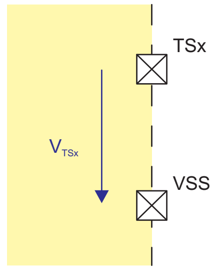

Figure 6. Temperature Calibration

2.6.1 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 = AAaa of *DAStatus2()*

     Is TINT > 0? If yes, TINT = AAaa – 2732.
4. Calculate temperature offset:

   $$\text{TINT offset} = \text{TEMP}_{TINT} - \text{TINT} + \text{TINT offset}_{old} \quad (6)$$
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.

2.6.2 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:

   $$\text{TSx offset} = \text{TEMP}_{TSx} - \text{TSx} + \text{TSx offset}_{old} \quad (7)$$

   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.