OVERVIEW

The P4 calibration flow charts describe the algorithms used to calibrate the PS401 Accuron™ battery manager. The PS401 A/D converter can be configured to perform certain measurements using an ADL (A/D converter List). This feature is only available in firmware revisions that are “ADL enabled”. All firmware revisions are ADL enabled for both voltage and current except those in the following table.

TABLE 1:  FIRMWARE REVISIONS WITHOUT ADL CAPABILITY

<table>
<thead>
<tr>
<th>No Voltage ADL</th>
<th>No Current ADL</th>
</tr>
</thead>
<tbody>
<tr>
<td>PS401-0107</td>
<td>PS401-0107</td>
</tr>
<tr>
<td>PS401-0108</td>
<td>PS401-0108</td>
</tr>
<tr>
<td>PS401-0109</td>
<td>PS401-0109</td>
</tr>
<tr>
<td>PS401-0202</td>
<td>PS401-0202</td>
</tr>
<tr>
<td>PS401-0203</td>
<td>PS401-0203</td>
</tr>
<tr>
<td>PS401-0204</td>
<td>PS401-0204</td>
</tr>
<tr>
<td>PS401-0301</td>
<td>PS401-0301</td>
</tr>
<tr>
<td>PS401-0302</td>
<td>PS401-0302</td>
</tr>
<tr>
<td></td>
<td>PS401-0303</td>
</tr>
<tr>
<td></td>
<td>PS401-0401</td>
</tr>
</tbody>
</table>

ADL capability is not essential to P4 calibration. The flow charts include information for calibrating all PS401 firmware revisions. The advantages of using the ADL list are quicker calibration (calibration delays are shortened) and improved current calibration accuracy. To set a particular ADL mode for calibration, write to the PS401 using a SMBus Block Write protocol:

data[0] = 0x01
number of bytes = 2
destination SMBus register is 0x24.

The write can be verified by reading RAM register 0xCD (one byte, bank 0/1). The returned value should be the same as the ADL mode number (0, 1 or 2) written to the part.

The part can be returned to ADL_NORMAL mode by again writing to the SMB register 0x24 using the SMB BLOCK WRITE protocol, but the data looks like:

data[0] = 0x00
number of bytes = 1

So, the procedure for using this mode in calibration is:
1. Change P4 to ADL mode of choice using SMB BLOCK WRITE protocol
2. Delay ~500 ms to allow part to update RAM
3. Verify change by reading RAM register
4. Perform calibration
5. Return P4 to Normal mode

The flow charts also use a term “SLEEP factor”, which is a wait state. The calibration procedure should wait before proceeding.
FIGURE 1: P4 CALIBRATION FLOW

- Attach pack or module
- Voltage Calibration
- Vcell Calibration
- Current Calibration
- Temperature Calibration
- Temperature Check
- T-Pin Check
- Exit
FIGURE 2: PACK VOLTAGE CALIBRATION

Start

Read "High", "Low", and "Dev" (deviation)

Read FLAGS1 and save value

Write new FLAGS1 (FLAGS1 AND 0xFE) to disable sample mode

Read value of CFVPack from P4

Firmware ADL enabled?

YES

SLEEP factor = 2
Force ADL_VOLTAGE mode

NO

SLEEP factor = 10
Current = 10 mA

BattVolt = Voltage (0x09) value from P4

PCVolt = voltage value from calibration hardware

Start loop

COMPUTE NEW CALIBRATION FACTOR(s)
CFVPack' = CFVPack * PCVolt / BattVolt
COVPack' = 0

WRITE NEW CALIBRATION FACTOR(s) TO DEVICE
EEPROM[CFVPack] = CFVPack'
EEPROM[COVPack] = COVPack'

Firmware ADL enabled?

NO

YES

Force ADL_VOLTAGE mode

delay = 550 ms * SLEEP factor

BattVolt = Voltage (0x09) value from P4

while |BattVolt - PCVolt| < "Dev"
FIGURE 2: PACK VOLTAGE CALIBRATION (Continued)

Firmware ADL enabled?

YES

Force ADL_NORMAL mode

NO

Current = 0 mA

Restore FLAGS1 saved value

End

FIGURE 3: CELL VOLTAGE CALIBRATION

Start

APPLY TEST VOLTAGES TO VCELL INPUTS

WAIT = 500ms * NCELLS (i.e. 1500ms)

assuming device is in "RUN" mode (not SAMPLE)

(ensure all cells are read)

READ VCELL VOLTAGE(s) FROM DEVICE

VCELL1 = SMBus Command Code 0x3C
VCELL2 = SMBus Command Code 0x3D
VCELL3 = SMBus Command Code 0x3E
VCELL4 = N/A

READ CALIBRATION FACTOR(s) FROM DEVICE

CF1 = EEPROM[CFVCell1]
CF2 = EEPROM[CFVCell2]
CF3 = EEPROM[CFVCell3]
CF4 = N/A

READ VOLTAGE(s) FROM CAL SYSTEM

V1 = Voltage at VCELL1 input
V2 = Voltage at VCELL2 input
V3 = Voltage at VCELL3 input
V4 = N/A

COMPUTE NEW CALIBRATION FACTOR(s)

CF1' = (V1 * CF1) / (VCELL1+VCELL2+VCELL3)
CF2' = (V2 * CF2) / (VCELL2+VCELL3)
CF3' = (V3 * CF3) / (VCELL3)
CF4' = N/A

WRITE NEW CALIBRATION FACTOR(s) TO DEVICE

EEPROM[CFVCell1] = CF1'
EEPROM[CFVCell2] = CF2'
EEPROM[CFVCell3] = CF3'

Pass - display

CFVCellx, VCellx, and Vx

End
FIGURE 4: CURRENT CALIBRATION

Start

Read "Test", "Cal", and "Dev" (deviation)

Start loop

Firmware ADL enabled?

YES

Force ADL_CURRENT_OFF_SET mode

delay = 100 ms * SLEEP factor

Read CFCurr from P4

Read offset (RAM 0xE7 value)

Firmware ADL enabled?

NO

SLEEP factor = 10

NO

COCurr = 0
COD = 0

Firmware ADL enabled?

YES

delay = 100 ms * SLEEP factor

BattCurr = Current (0x0A) value from P4

YES

Write CFCurr = 16384 (Current (0x0A) returns raw A/D value)

SLEEP factor = 5
Force ADL_CURRENT mode

COD = BattCurr - offset
COCurr = BattCurr

COD = BattCurr * 16384 / CFCurr
COCurr = COD

COD = BattCurr
COCurr = COD

NO

delay = 100 ms * SLEEP factor

BattCurr = Current (0x0A) value from P4
FIGURE 4: CURRENT CALIBRATION (Continued)

- Firmware ADL enabled?
  - NO
    - Force ADL_CURRENT mode
    - Apply "Cal" current
    - delay = 200 ms * SLEEP factor
  - YES
    - Write COD to P4
    - Write COCurr to P4
    - Firmware ADL enabled?
      - NO
      - WRITE NEW CALIBRATION FACTOR
        - EEPROM[CFCurr] = CFCurr'
        - Firmware ADL enabled?
          - YES
          - Force ADL_CURRENT mode
            - "Test" current = "Cal" current?
              - YES
              - Apply "Test" current
              - delay = 1000 ms
              - WRITE NEW CALIBRATION FACTOR
                - EEPROM[CFCurr] = CFCurr'
                - Firmware ADL enabled?
                  - YES
                  - Force ADL_CURRENT mode
                    - Firmware ADL enabled?
                      - NO
                      - Firmware ADL enabled?
FIGURE 4: CURRENT CALIBRATION (Continued)

- Firmware ADL enabled?
  - NO
  - Force ADL_NORMAL mode
  - End
  - YES

- Remove "Test" current

- while |BattCurr - PCCurr| < "Dev"

- BattCurr = Current (0x0A) value from P4

- PCCurr = Current value from calibration hardware

- delay = 100 ms * SLEEP factor
FIGURE 5: TEMPERATURE CALIBRATION

Start

Read “Dev” (deviation)

Read Temperature (0x08) value from P4

Read CALTEMP value from calibration hardware

Read value of CFTempI from P4

|CALTEMP - Temperature| < “Dev” ?

Calculation:

\[ \text{CFTempI'} = \frac{\text{old CFTempI} \times \text{CALTEMP}}{\text{Temperature}} \]

YES

Write CFTempI’ to P4

Delay = 300 ms

NO

Pass - display CFTempI’, CALTEMP, and Temperature

End

Note: All temperature values in Kelvin

Delay = 300 ms

Read Temperature (0x08) value from P4

Read CALTEMP value from calibration hardware
FIGURE 6: TEMPERATURE CHECK

Start

Read “High”, “Low”, and “Dev” (deviation)

Read Temperature (0x08) value from P4

Read CALTEMP value from calibration hardware

Temperature < "High" ?

Temperature > "Low" ?

|CALTEMP - Temperature| < "Dev" ?

Pass - display CALTEMP, and Temperature

End

Fail, Error

Fail, Error

NO

NO

NO

YES

YES

YES
FIGURE 7: T-PIN CHECK

Start

Read "High", "Low"

Read $I_{\text{cal_tpin}}$ value from calibration hardware

\[
\text{tpin\_value (ohms)} = \left(\frac{10000 \times (10 \times I_{\text{cal_tpin}}/32768)}{7.5 - (10 \times I_{\text{cal_tpin}}/32768)}\right)
\]

\begin{center}
\begin{tabular}{c}
\text{tpin\_value < "High" ?} \\
\text{tpin\_value > "Low" ?} \\
\text{Pass - display tpin\_value} \\
\text{End}
\end{tabular}
\end{center}

Fail, Error
Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip’s products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

**Trademarks**

The Microchip name and logo, the Microchip logo and PowerSmart are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

Accuron, PowerTool, SmartSensor, SmartShunt, SmartTel, PowerCal and PowerInfo are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

All other trademarks mentioned herein are property of their respective companies.


*Printed on recycled paper.*