ACKNOWLEDGEMENT

By utilizing this website and/or documentation, I hereby acknowledge as follows:

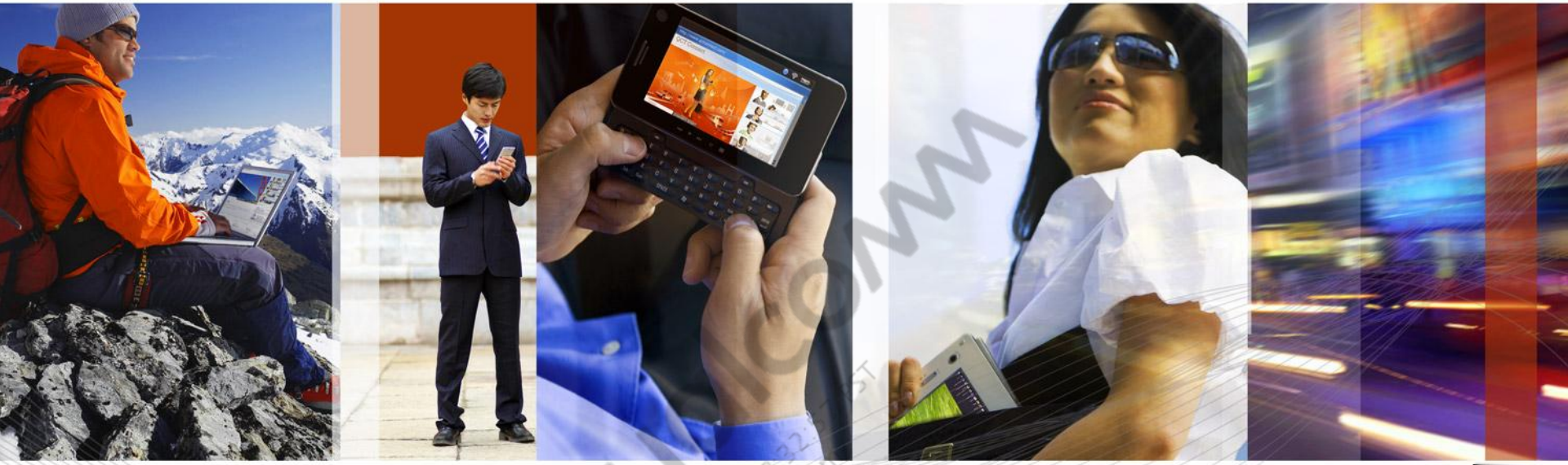
Effective October 1, 2012, QUALCOMM Incorporated completed a corporate reorganization in which the assets of certain of its businesses and groups, as well as the stock of certain of its direct and indirect subsidiaries, were contributed to Qualcomm Technologies, Inc. (QTI), a wholly-owned subsidiary of QUALCOMM Incorporated that was created for purposes of the reorganization.

Qualcomm Technology Licensing (QTL), the Company's patent licensing business, continues to be operated by QUALCOMM Incorporated, which continues to own the vast majority of the Company's patent portfolio. Substantially all of the Company's products and services businesses, including QCT, as well as substantially all of the Company's engineering, research and development functions, are now operated by QTI and its direct and indirect subsidiaries[1]. Neither QTI nor any of its subsidiaries has any right, power or authority to grant any licenses or other rights under or to any patents owned by QUALCOMM Incorporated.

No use of this website and/or documentation, including but not limited to the downloading of any software, programs, manuals or other materials of any kind or nature whatsoever, and no purchase or use of any products or services, grants any licenses or other rights, of any kind or nature whatsoever, under or to any patents owned by QUALCOMM Incorporated or any of its subsidiaries. A separate patent license or other similar patent-related agreement from QUALCOMM Incorporated is needed to make, have made, use, sell, import and dispose of any products or services that would infringe any patent owned by QUALCOMM Incorporated in the absence of the grant by QUALCOMM Incorporated of a patent license or other applicable rights under such patent.

Any copyright notice referencing QUALCOMM Incorporated, Qualcomm Incorporated, QUALCOMM Inc., Qualcomm Inc., Qualcomm or similar designation, and which is associated with any of the products or services businesses or the engineering, research or development groups which are now operated by QTI and its direct and indirect subsidiaries, should properly reference, and shall be read to reference, QTI.

[1] The products and services businesses, and the engineering, research and development groups, which are now operated by QTI and its subsidiaries include, but are not limited to, QCT, Qualcomm Mobile & Computing (QMC), Qualcomm Atheros (QCA), Qualcomm Internet Services (QIS), Qualcomm Government Technologies (QGOV), Corporate Research & Development, Qualcomm Corporate Engineering Services (QCES), Office of the Chief Technology Officer (OCTO), Office of the Chief Scientist (OCS), Corporate Technical Advisory Group, Global Market Development (GMD), Global Business Operations (GBO), Qualcomm Ventures, Qualcomm Life (QLife), Quest, Qualcomm Labs (QLabs), Snaptracs/QCS, Firethorn, Qualcomm MEMS Technologies (QMT), Pixtronix, Qualcomm Innovation Center (QuIC), Qualcomm iSkoot, Qualcomm Poole and Xiam.



QUALCOMM CDMA Technologies

REDEFINING MOBILITY

Battery Monitoring System (BMS) Training

Application Note

80-VB073-17 Rev. B

Restricted Distribution. Not to be distributed to anyone who is not an employee of either Qualcomm or a subsidiary of Qualcomm without the express approval of Qualcomm's Configuration Management.

Not to be used, copied, reproduced in whole or in part, nor its contents revealed in any manner to others without the express written permission of Qualcomm.

QUALCOMM is a registered trademark of QUALCOMM Incorporated in the United States and may be registered in other countries. Other product and brand names may be trademarks or registered trademarks of their respective owners. CDMA2000 is a registered certification mark of the Telecommunications Industry Association, used under license. ARM is a registered trademark of ARM Limited. QDSP is a registered trademark of QUALCOMM Incorporated in the United States and other countries.

This technical data may be subject to U.S. and international export, re-export, or transfer ("export") laws. Diversion contrary to U.S. and international law is strictly prohibited.

QUALCOMM Incorporated
5775 Morehouse Drive
San Diego, CA 92121-1714
U.S.A.

Revision History

Revision	Date	Description
A	June 08, 2012	Initial release
B	June 29, 2012	Added the following slides: ▪ Slide 23: Advanced Hardware Topics – State Transition Details ▪ Slide 25: Advanced Hardware Topics – S1 and S2 State Timing ▪ Slide 26: Advanced Hardware Topics – S3 State Timing Updated the following slides: ▪ Slide 14: BMS Algorithms – SoC Calculation ▪ Slide 17: Battery Characterization ▪ Slide 18: Battery Characterization (cont.) ▪ Slide 24: Advanced Hardware Topics – State Transition Timing ▪ Slide 44: Advanced BMS Algorithms – Dynamic UUC ▪ Slide 45: Advanced BMS Algorithms – Dynamic UUC (cont.)

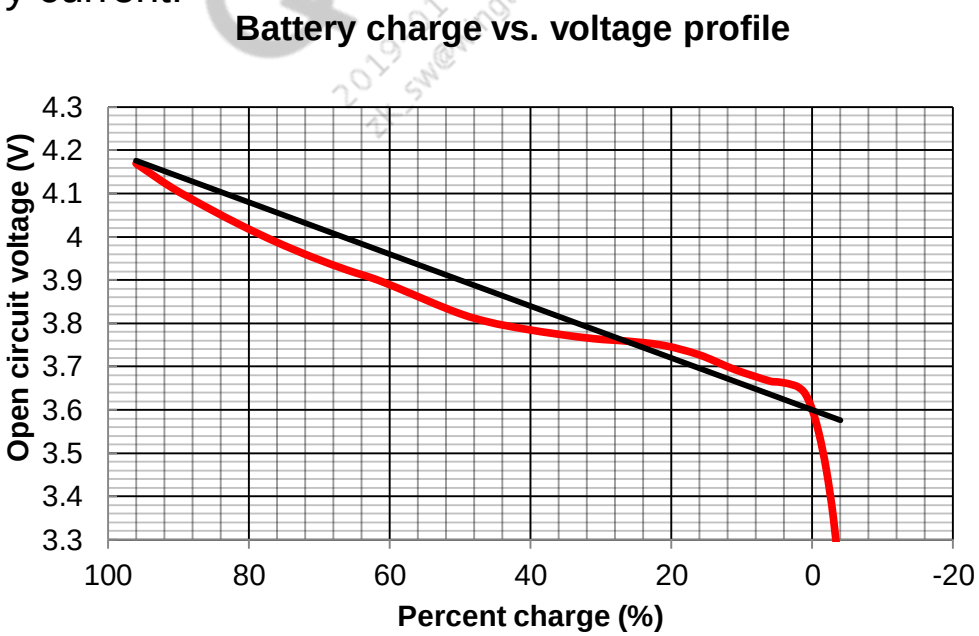
Contents

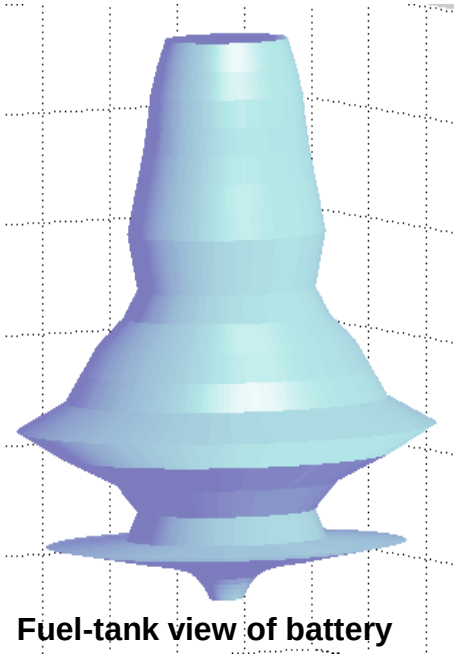
High-level BMS Overview

REDEFINING MOBILITY

What is a Fuel Gauge?

- In general, a fuel gauge is an instrument used to indicate the level of the fuel in a tank.
  - In this case, the fuel is representative of the battery life.
- It is critical that the remaining battery life is reported to the end user.
- The battery monitoring system (BMS) provides the necessary functions to:
  - Calculate the state of charge (SoC) or remaining battery life based on an algorithmic approach.
  - Use battery voltages, including open circuit voltage (OCV), when applicable.
  - Monitor battery current.





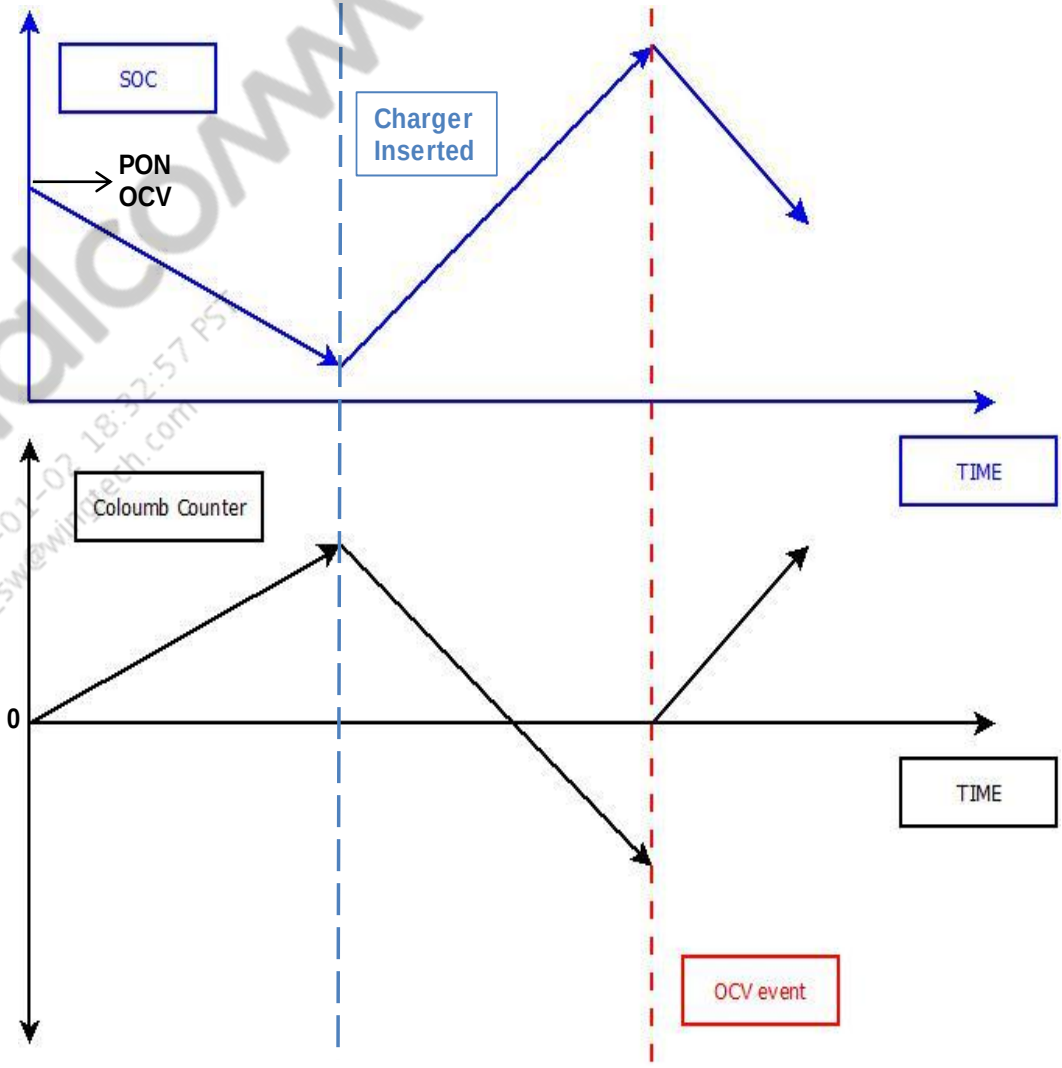
Fuel-tank view of battery

BMS Acronyms and Definitions

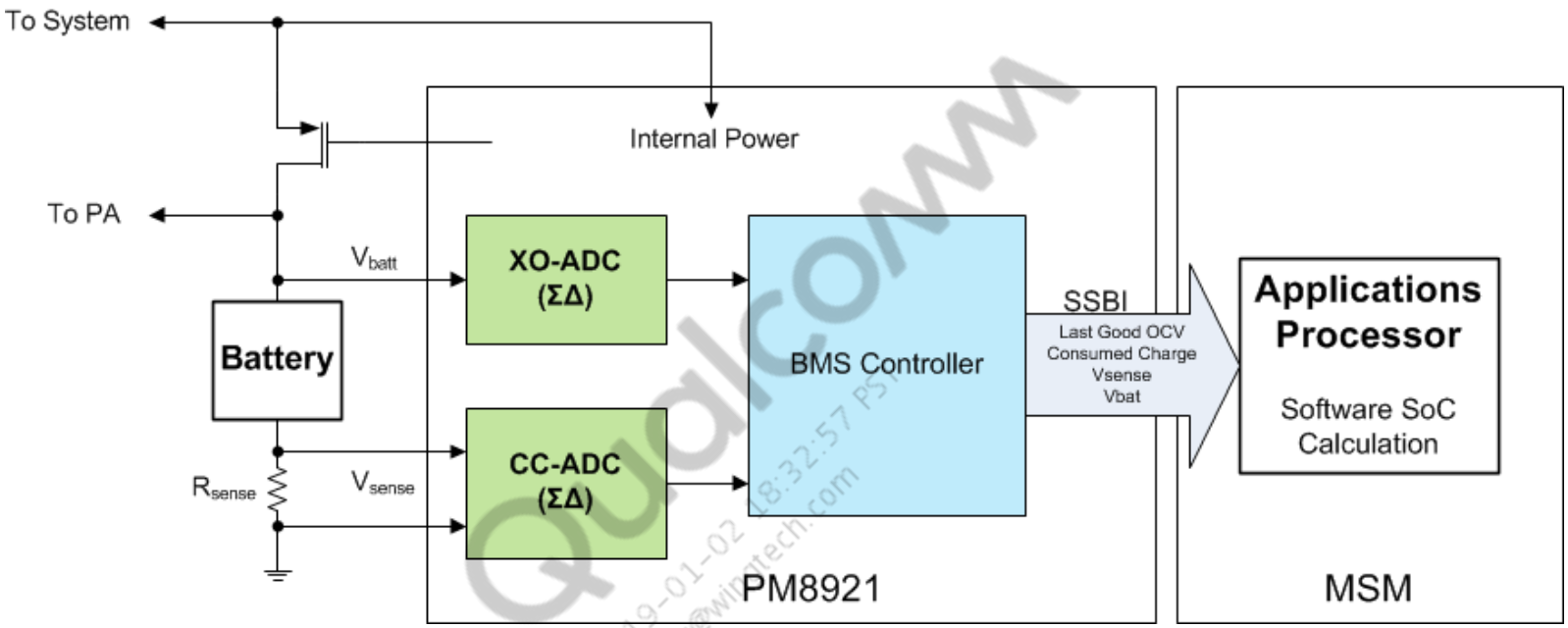
Acronym	Description
BMS	**Battery monitoring system** – A system that reports SoC.
FCC	**Full-charge capacity** – The total amount of charge that can be extracted from a fully charged battery. FCC is defined as the amount that can be extracted from the battery at a discharge current less than $^{1}/_{20}$C. FCC changes with age and cycle life of the battery.
RC	**Remaining capacity** – The amount of charge remaining in the battery at its current state. After a full charge, RC = FCC. Like FCC, it is assumed that the discharge current is less than $^{1}/_{20}$C.
UUC	**Unusable capacity** – The battery capacity that cannot be used due to the voltage drop across the battery impedance, reducing the battery voltage below the failure voltage. It is a function of discharging current.
RUC	**Remaining usable capacity** -- The remaining capacity (RC) minus the unusable capacity (UUC). RUC = RC - UUC.
SoC	**State of charge** – The ratio of the remaining capacity to the total capacity (SoC = RC/FCC). When reported to the end user, it is useful to include UUC: SoC = RUC/UC = (RC - UUC) / (FCC - UUC).
C (rate)	A measurement of the discharge rate at which the battery would be depleted in 1 hour (i.e., a battery rated at 1 Ah provides 1 A for one hour, if discharged at 1C). The same battery discharged at 0.5C would provide 500 mA for two hours.
OCV	**Open circuit voltage** – Battery voltage at steady-state near-zero (generally, less than C/20) current. (Note that battery voltage takes 5 to 30 minutes to settle to OCV after a load).
Rbatt	**Battery resistance –** Internal resistance of the battery.

BMS Hardware Architecture – General

- The BMS provides accurate battery SoC.
- SoC is primarily based on battery-specific OCV vs. SoC curve.
  - Good battery OCV vs. SoC profile is necessary for accurate SoC calculation.
  - OCV vs. SoC curve is stored as a lookup table.
  - Poweron OCV occurs at boot, and is used as the starting SoC point.
  - Periodic OCV measurements are used to generate SoC.
- Coulomb counting is used for SoC updates between OCV measurements.
- Autonomous BMS hardware gathers raw OCV and Coulomb count (CC) data, which software will use to calculate the SoC.



BMS Hardware Architecture – Block Diagram



- CCADC samples the V_{sense} generated across a 10 to 25 mΩ sense resistor.
  - The V_{sense} measurement will translate to a current measurement (Coulomb counting).
- HKADC samples Vbat, which translates to an OCV measurement.
  - Shared use with other housekeeping functions
- The BMS controller autonomously manages ADCs and key measurements.
  - Controls measurement frequency, averaging, Coulomb counting, and CC resets.
  - All data for SoC calculations is stored, read as desired by software → no periodic software intervention.
  - The BMS controller uses a Qualcomm proprietary algorithm.
- PMIC software reads stored data and executes software algorithm to perform SoC calculation.

BMS Hardware Architecture – BMS Controller Functional Details

- The BMS controller is a finite state machine, which functions automatically and reliably without software control to:
  - Minimize power consumption without impacting accuracy.
  - Determine when battery voltage has settled sufficiently to use V_{batt} as a reliable indicator of SoC, and do so.
  - Accept an input consisting of occasional high-magnitude current spikes during standby (GSM paging and similar), without wasting power or disrupting measurements (details discussed in later slides).
  - Handle different (and potentially unknown) battery chemistries and unusual discharge profiles (such as multimedia mode).
  - Store all raw data for SoC calculation, to be read by software as needed.

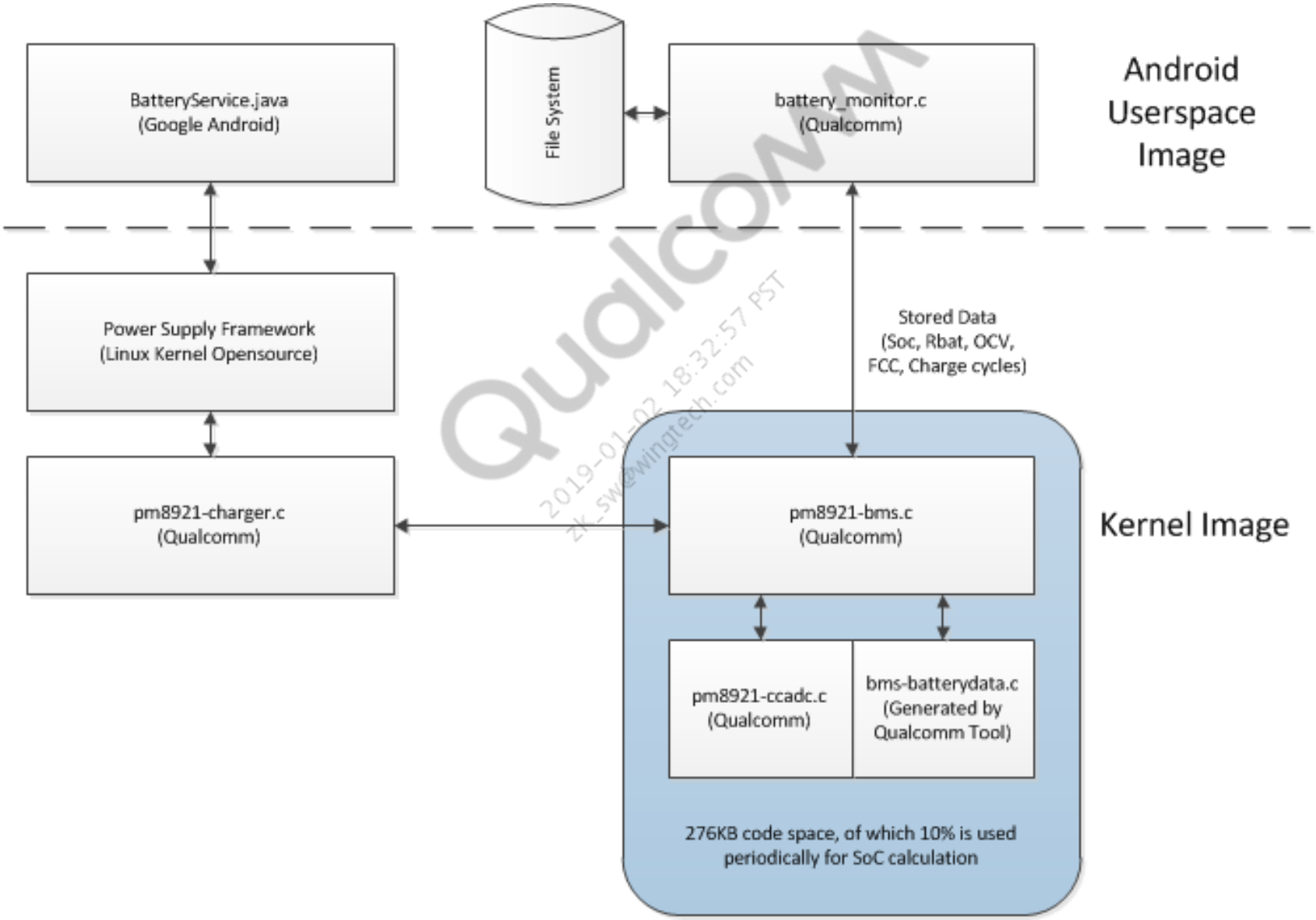
BMS Software Architecture – General

- Role of application processor software:
  - Calculate SoC using raw data from BMS controller.
    - Execute SoC calculation when requested by host.

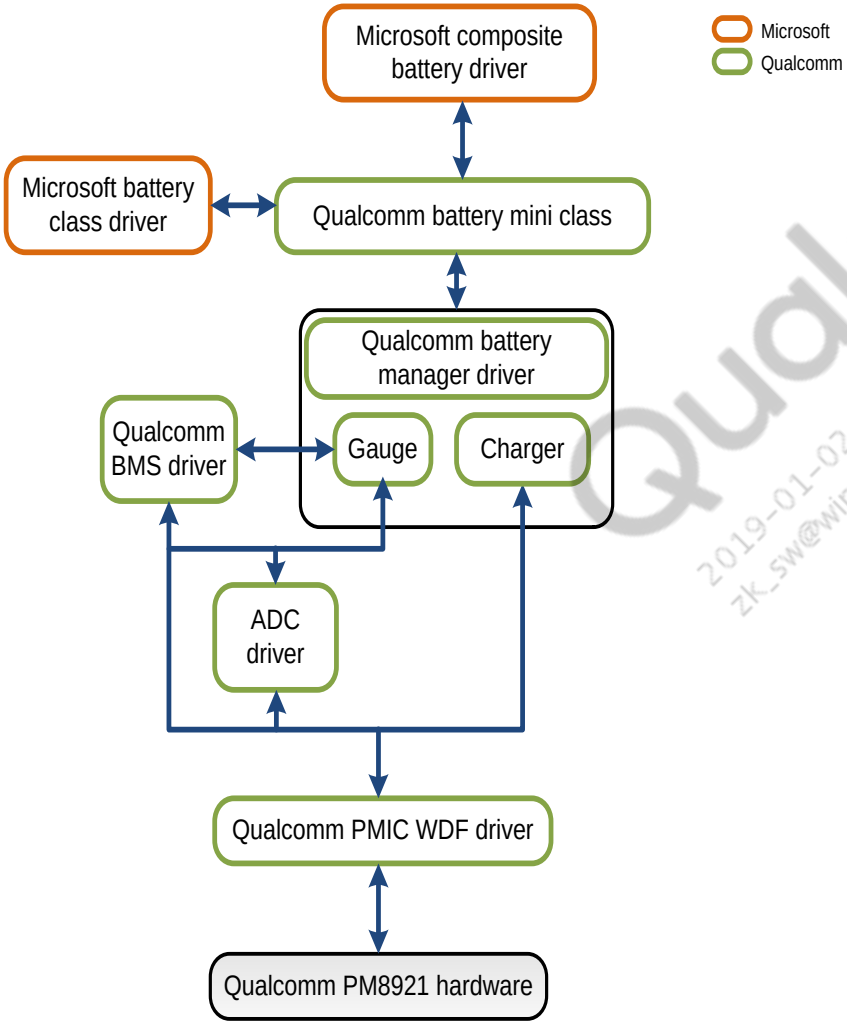
    or

    - Periodically calculate SoC, and notify host at specified SoC thresholds.
  - Perform temperature corrections based on V_{therm} data.
  - Store and use battery-specific data:
    - OCV vs. SoC curve
    - Battery resistance
    - Battery aging (if aging data is embedded in the battery profile)
    - Temperature coefficients
    - Program BMS settings
  - Handle *special cases* (learning cycle, initialization)

BMS Software Architecture – Linux Android



BMS Software Architecture – Windows Mobile



- Qualcomm battery mini-class
  - qcbattminiclass8960.sys
  - Interface Qualcomm battery manager driver with MSFT kernel battery drivers.
- Qualcomm battery manager driver
  - qcbattmngr8960.sys
  - Manage charging and gauging activities for Windows HLOS.
- Qualcomm BMS driver
  - qcbms8960.sys
  - **Implement main BMS functionality**
- ADC driver
  - qcxoadc8960.sys
  - Provide PMIC ADC readings to battery manager driver for monitoring purpose.
- Qualcomm PMIC WDF driver
  - qcpmic8960.sys
  - Low level PMIC resource access

BMS Algorithms – SoC Calculation

- Hybrid BMS architecture is based on OCV values:
  - OCV values generate SoC by being mapped to the battery profile table lookup.
  - OCV is used as a starting point at system boot.
  - OCV is used to update SoC and to reset the Coulomb counter.
- Coulomb count (CC) keeps track of lost charge between OCV updates.
- FCC is obtained from the battery profile table.
  - BMS uses unloaded, or OCV-based FCC values.
  - The unloaded FCC table is fairly flat over temperature → loaded FCC table is not!
- BMS SoC is calculated as follows:

$$SoC = \frac{\text{Lookup}(\boldsymbol{OCV}) * \text{FCC} - \frac{\boldsymbol{CC}}{R_{sense}} - UUC}{FCC - UUC}$$

$$CC/R_{sense}(\text{mAh}) = CC(\text{decimal}) * \frac{86.8056\ \mu\text{V} * 0.0016785\ \text{s}}{3600\ \text{s/hr} * R_{sense}}$$

- Example for 4.2 V rated battery if poweron OCV is 4.2 V:

$$SoC = \frac{100\% * \text{FCC} - \frac{\boldsymbol{CC}}{R_{sense}} - UUC}{FCC - UUC} = \frac{\text{FCC} - UUC}{FCC - UUC} = 100\%$$

BMS Algorithms – UUC Calculation

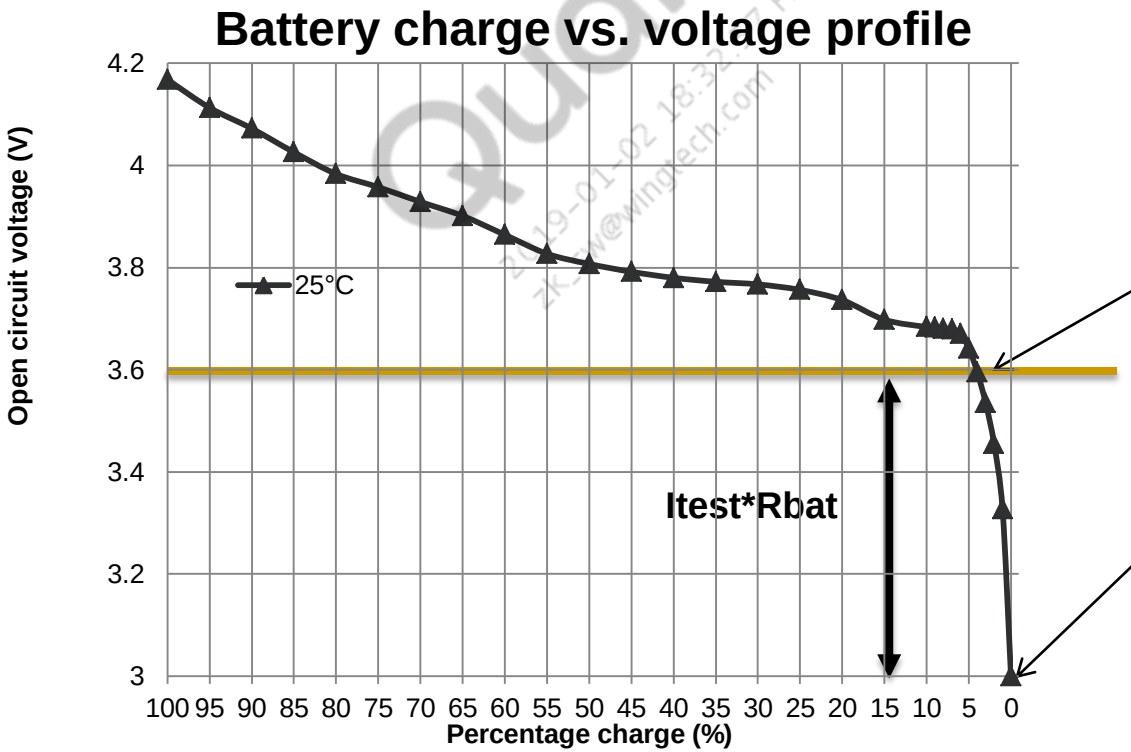
- BMS compensates SoC with UUC.
- UUC adds temperature dependency to SoC with unloaded FCC tables.
- UUC changes with temperature due to battery resistance changes.
  - UUC calculates stored charge that is made inaccessible due to the voltage drop across Rbatt, pushing Vbatt to the system cutoff voltage → Vfailure.
  - UUC uses the long-term value of battery resistance (settled value).
  - UUC is approximated as follows:

$$UUC = FCC * Lookup(R_{batt} * I_{test} + V_{failure})$$

  - Where:
    - FCC is full charge capacity of the battery.
    - Lookup() refers to the battery-profile SoC table-lookup value.
    - R_{batt} is the battery resistance as a function of temperature.
    - I_{test} is the desired peak system current that is being protected against.
    - $V_{failure}$ is the characterized unloaded battery cutoff voltage (based on the battery profile).

BMS Algorithms – Battery Profile and UUC Value

- $V_{failure}$ is the characterized unloaded battery cutoff voltage.
- The following battery profile has $V_{failure}$ = 3.0 V open-circuit voltage (no load).
- UUC shifts the SoC = 0% point to compensate for I*R drop.
- If the 25 C Rbat = 0.2 Ω near the end of the curve, and the worst-case current is Itest = 3.0 A, then UUC will shift the unloaded SoC = 0% point to 3.6 V.
- Therefore, applying UUC allows for the cutoff voltage to be at $V_{failure}$ under load.



UUC Values

Battery Resistance (Ω)	Itest (A)	IR (V)	SoC Shift (%)
0.2	3.0	0.6	4

Itest shifts the SoC = 0% point to 3.6 V.

Vfailure = 3.0 V, defines the initial SoC = 0% point.

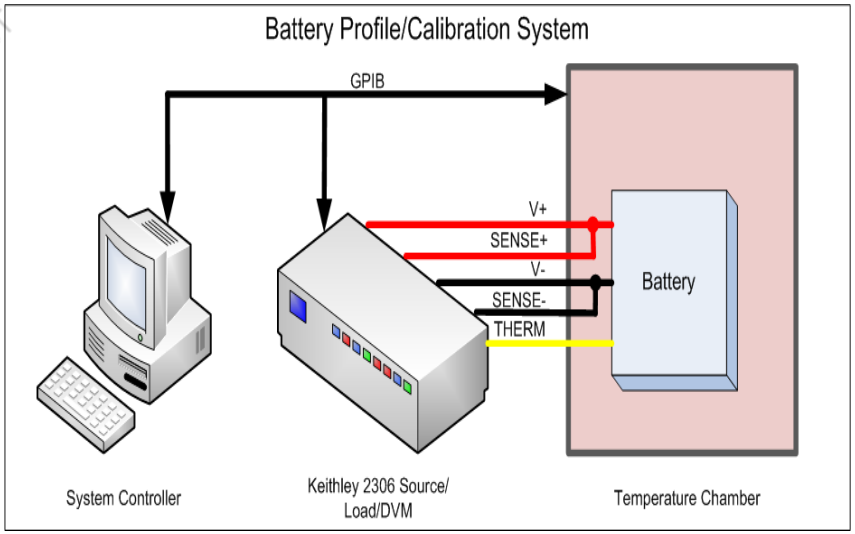
Battery Characterization

- Battery characterization profile is needed for the BMS to operate; details can be found in *Application Note: Battery Characterization Test Procedure* (80-VA360-12).
- Qualcomm provides the battery data-collection software tool (QDART/QSPR), starting with QDART version 4.7.20.
  - Automate data collection based on battery specifications.
  - Generate data tables based on tool inputs.
  - Load data tables in software.
- Uses Keithley 2306/2420 or Agilent N6705 as source/load → accurate over current range.
- Supports many temperature chambers → refer to the application note above.
- The tool can scale data collection to meet customer needs.
  - Higher discharge current (faster run time, poorer data)
  - Enable charge-cycle aging, if needed.
  - Temperature compensation (65 C, 40 C, 25 C, 0 C, and -20 C, by default)
- Tool provides the following data:
  - OCV across temperature and SoC
  - FCC across temperature
  - Rbatt across temperature and SoC
    - **Rbatt increases dramatically at cold temperature and <10% SoC.**

Battery Characterization *(cont.)*

OCV and FCC example table

		Temperature (ºC)						Charge cycles - aging scale factor				
		-20	0	25	40	60		100	200	300	400	500
FCC		1468	1476	1478	1478	1480	FCC scale factor	0.9	0.84	0.81	0.78	0.74
OCV @ percent charge	100	4166	4166	4159	4156	4152	OCV scale factor @ percent charge 100	0.99	0.95	0.92	0.88	0.87
	95	4096	4106	4106	4105	4103	95	0.99	0.95	0.92	0.88	0.87
	90	4043	4061	4062	4061	4060	90	0.99	0.95	0.92	0.88	0.87
	85	3983	4019	4022	4022	4020	85	0.99	0.95	0.92	0.88	0.87
	80	3939	3981	3986	3986	3984	80	0.99	0.95	0.92	0.88	0.87
	75	3902	3946	3954	3953	3951	75	0.99	0.95	0.92	0.88	0.87
	70	3871	3908	3925	3925	3922	70	0.99	0.95	0.92	0.88	0.87
	65	3845	3871	3896	3898	3895	65	0.99	0.95	0.92	0.88	0.87
	60	3825	3842	3858	3867	3866	60	0.99	0.95	0.92	0.88	0.87
	55	3808	3818	3826	3828	3827	55	0.99	0.95	0.92	0.88	0.87
	50	3792	3798	3806	3808	3808	50	0.99	0.95	0.92	0.88	0.87
	45	3778	3786	3791	3794	3793	45	0.99	0.95	0.92	0.88	0.87
	40	3764	3777	3779	3782	3781	40	0.99	0.95	0.92	0.88	0.87
	35	3751	3769	3771	3772	3770	35	0.99	0.95	0.92	0.88	0.87
	30	3739	3758	3765	3763	3755	30	0.99	0.95	0.92	0.88	0.87
	25	3726	3742	3754	3751	3736	25	0.99	0.95	0.92	0.88	0.87
	20	3712	3717	3733	3728	3712	20	0.99	0.95	0.92	0.88	0.87
	15	3691	3696	3698	3696	3681	15	0.99	0.95	0.92	0.88	0.87
	10	3653	3685	3677	3676	3667	10	0.99	0.95	0.92	0.88	0.87
	9	3643	3681	3675	3675	3665	9	0.99	0.95	0.92	0.88	0.87
	8	3629	3676	3673	3674	3663	8	0.99	0.95	0.92	0.88	0.87
	7	3612	3669	3670	3671	3653	7	0.99	0.95	0.92	0.88	0.87
	6	3590	3656	3662	3661	3628	6	0.99	0.95	0.92	0.88	0.87
	5	3562	3633	3635	3632	3584	5	0.99	0.95	0.92	0.88	0.87
	4	3526	3595	3587	3583	3528	4	0.99	0.95	0.92	0.88	0.87
	3	3477	3538	3524	3522	3455	3	0.99	0.95	0.92	0.88	0.87
	2	3404	3454	3440	3438	3349	2	0.99	0.95	0.92	0.88	0.87
	1	3282	3319	3307	3300	3207	1	0.99	0.95	0.92	0.88	0.87
	0	3000	3000	3000	3000	3000	0	0.99	0.95	0.92	0.88	0.87



Note: Remote sensing at battery terminals is critical for proper battery profile!

Battery Characterization – Battery Resistance

Rbatt example table

	Temperature				
SoC	-20 C	0 C	25 C	40 C	65 C
100	0.84298	0.44263	**0.236197**	0.215942	0.214117
95	0.94554	0.49085	0.248532	0.222819	0.221147
90	0.92173	0.48079	0.249395	0.225062	0.227080
85	0.92305	0.47580	0.255307	0.230862	0.232353
80	0.92449	0.47828	0.260792	0.232518	0.237202
75	0.92161	0.47257	0.260274	0.234272	0.240166
70	0.91899	0.47222	0.259099	0.234642	0.242015
65	0.92888	0.47721	0.238426	0.218495	0.236624
60	0.96147	0.48390	0.234259	0.211324	0.221455
55	1.01030	0.49046	0.235595	0.214144	0.225604
50	1.07485	0.50150	0.240061	0.216411	0.232379
45	1.16876	0.52065	0.245032	0.219034	0.237500
40	1.32600	0.54727	0.253491	0.224555	0.239852
35	1.49863	0.57903	0.263967	0.231072	0.230419
30	1.68581	0.60876	0.269035	0.231694	0.231639
25	1.86872	0.62795	0.268676	0.228942	0.235372
20	2.05704	0.68369	0.255198	0.223368	0.228962
15	2.29840	0.80346	0.292559	0.256219	0.246833
10	1.15449	0.56925	0.258565	0.227927	0.234642
9	1.20784	0.57994	0.260138	0.226310	0.233816
8	1.26229	0.59576	0.262098	0.224692	0.230676
7	1.36773	0.62122	0.263897	0.225638	0.227541
6	1.50153	0.65221	0.261260	0.223475	0.227971
5	1.72460	0.69446	0.257158	0.226042	0.233086
4	2.04931	0.77424	0.265620	0.230452	0.245917
3	2.57101	0.88246	0.280246	0.238887	0.270807
2	3.68272	1.07999	0.302108	0.248421	0.504143
1	30.4359	2.42318	1.50500	0.996750	7.72222
0	403.657	300.469	233.758	209.995	182.112

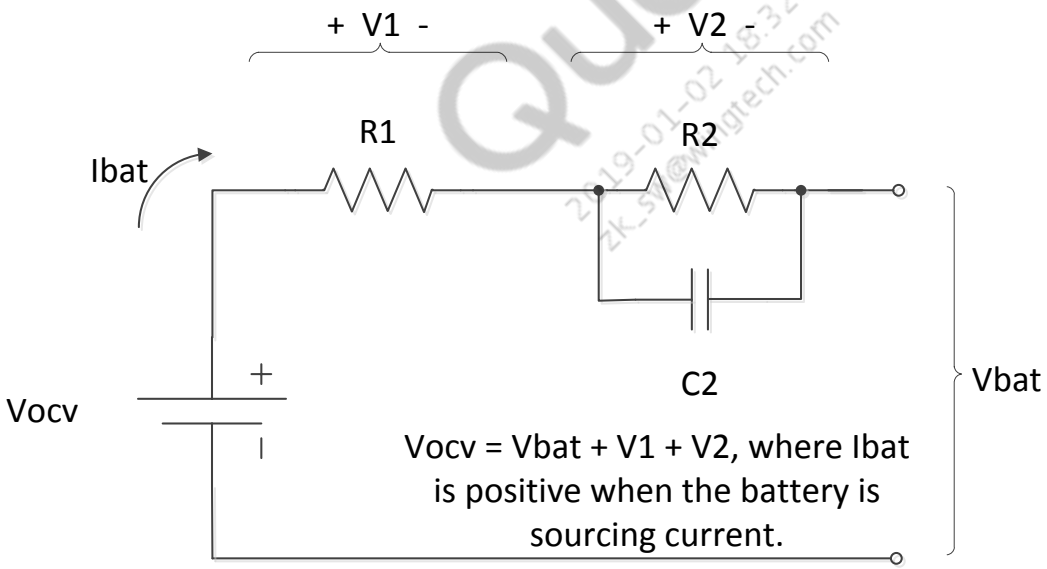
Rbatt scale-factor example table

Default Rbatt	0.236

	Temperature				
SoC	-20 C	0 C	25 C	40 C	65 C
100	357	187	**100**	91	91
95	400	208	105	94	94
90	390	204	106	95	96
85	391	201	108	98	98
80	391	202	110	98	100
75	390	200	110	99	102
70	389	200	110	99	102
65	393	202	101	93	100
60	407	205	99	89	94
55	428	208	100	91	96
50	455	212	102	92	98
45	495	220	104	93	101
40	561	232	107	95	102
35	634	245	112	98	98
30	714	258	114	98	98
25	791	266	114	97	100
20	871	289	108	95	97
15	973	340	124	108	105
10	489	241	109	96	99
9	511	246	110	96	99
8	534	252	111	95	98
7	579	263	112	96	96
6	636	276	111	95	97
5	730	294	109	96	99
4	868	328	112	98	104
3	1089	374	119	101	115
2	1559	457	128	105	213
1	12886	1026	637	422	3269
0	170899	127211	98968	88907	77102

Battery Characterization – Current Battery Model

- Current battery model
  - The battery profiling tool now provides a single value for resistance.
    - Rbatt = R1 + R2.
  - Current profiling method does not capture individual resistance values, and is being improved (as described later in this application note).



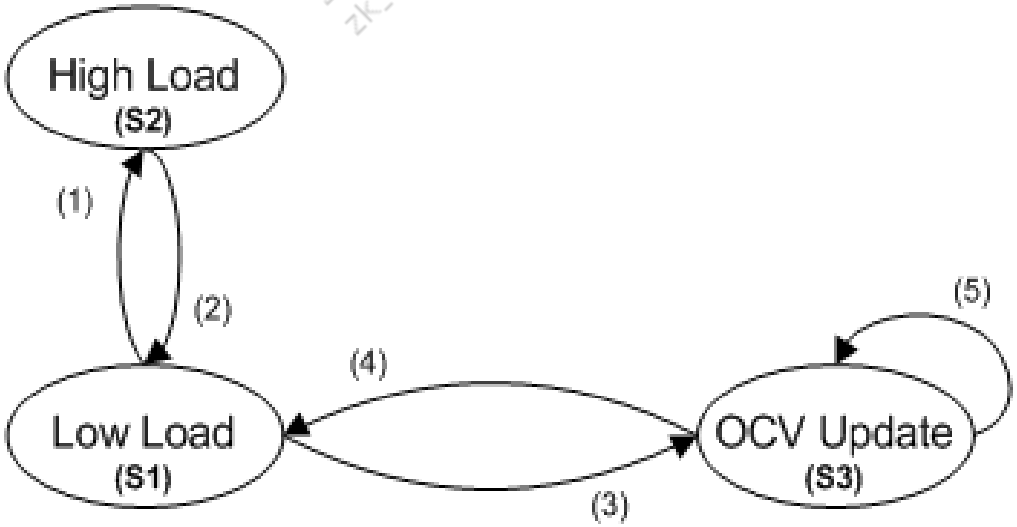
Current battery model

Advanced BMS Overview

REDEFINING MOBILITY

Advanced Hardware Topics – BMS State Machine Details

State	State details
Low-load state (S1)	• Reduces BMS power consumption by decreasing measurement frequency during phone standby.
High-load state (S2)	• Increases Coulomb counting accuracy with frequent measurements during active phone use.
OCV update (S3)	• Requests simultaneous Vbatt and Vsense measurements with a large number of samples. • The process OCV block evaluates Vbatt and, if appropriate, updates the last good OCV output and resets the Coulomb counter to zero.



Advanced Hardware Topics – State Transition Details

- State transitions
  - BMS counters dictate state transitions.
  - Counters are based on current thresholds in each state.
  - Successive counts are not required → The counter that overflows first determines the state.
  - Counters reset at each state transition.
- OCV qualifying measurement requirements
  - OCV measurement is attempted after 128 counts in low-load state (S1).
  - Simultaneous voltage and current measurements are taken.
  - Current must be less than OCV threshold (8 mA) during simultaneous measurement.
  - If current is greater than OCV threshold (8 mA), BMS will retry a maximum of four times.
  - After four attempts, BMS jumps back to S1.
- OCV used by system
  - Two qualifying OCV measurements must occur.
  - The two OCV measurements must be within the settled battery tolerance (1 mV).
  - The OCV value is stored in the BMS register, and the Coulomb counter gets reset.

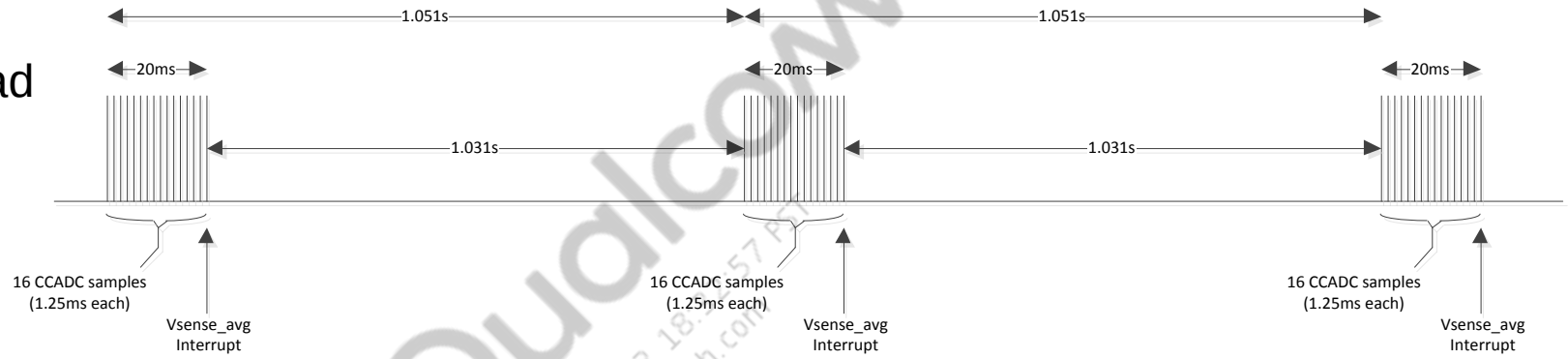
Advanced Hardware Topics – State Transition Timing

State transition timing							
State transition	**Sample delay (s)**	**Sample average**	**Conversion time (s)**	**Count required**	**Transition time (s)**	**Average current (µA)**	**Count-threshold requirement**
S1 -> S3	1.03215	16	0.01965	128	134.630	13	Current 0 to 50 mA to get into S3, and < 8 mA during actual OCV measurement.
S1 -> S2	1.03215	16	0.01965	8	8.414	13	Current > 50 mA.
S2 -> S1	0	8	0.00982	32	0.314	700	Current < 45 mA.
S3 -> S1	N/A	64	0.07859	4	0.314	1100	The state will transition in less than 4 counts if a qualifying measurement occurs sooner (current < 8 mA).

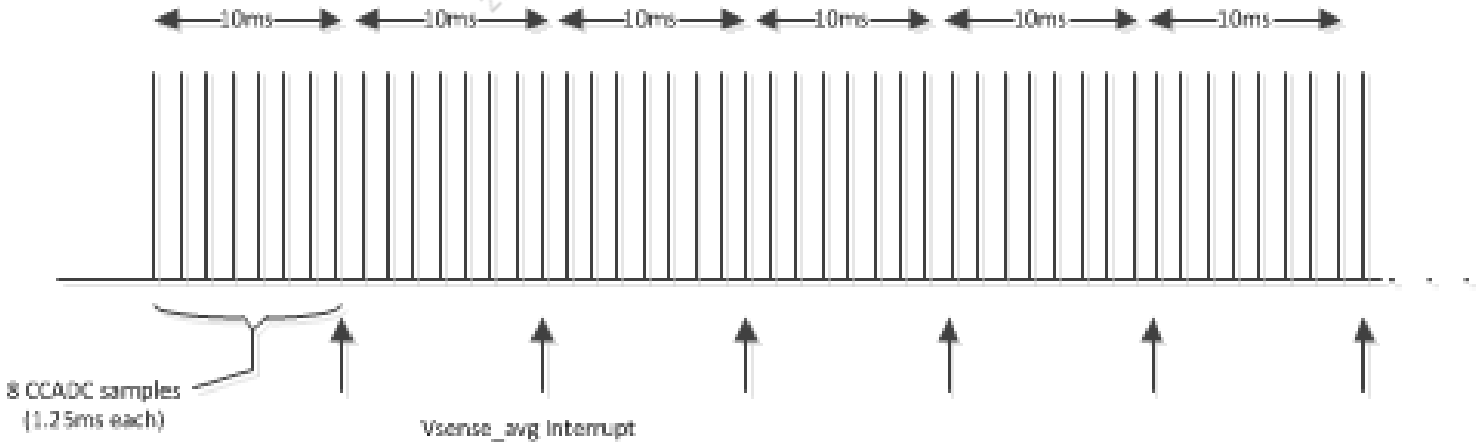
Advanced Hardware Topics – S1 and S2 State Timing

- State timing examples:

Low-load (S1)

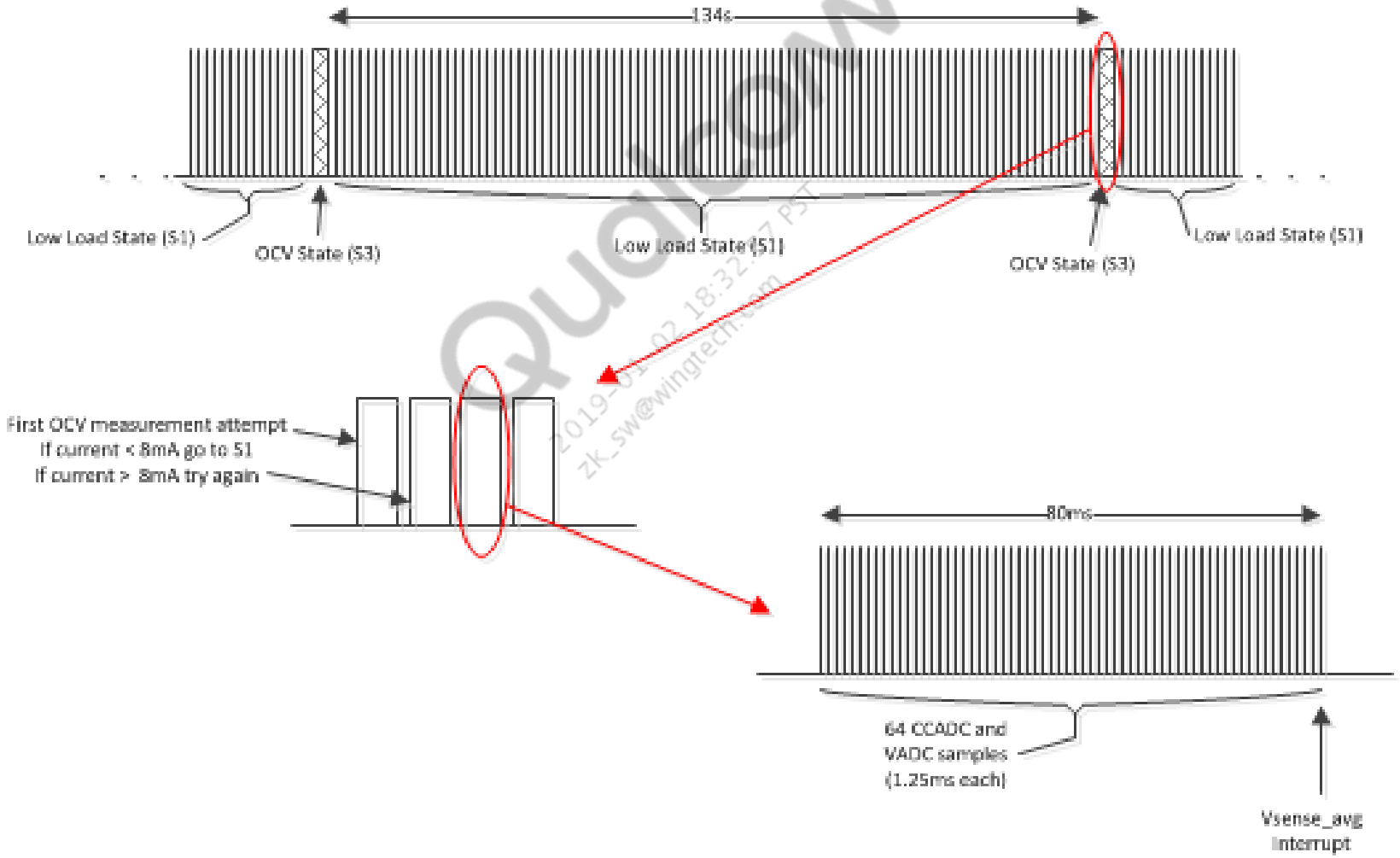


High-load (S2)



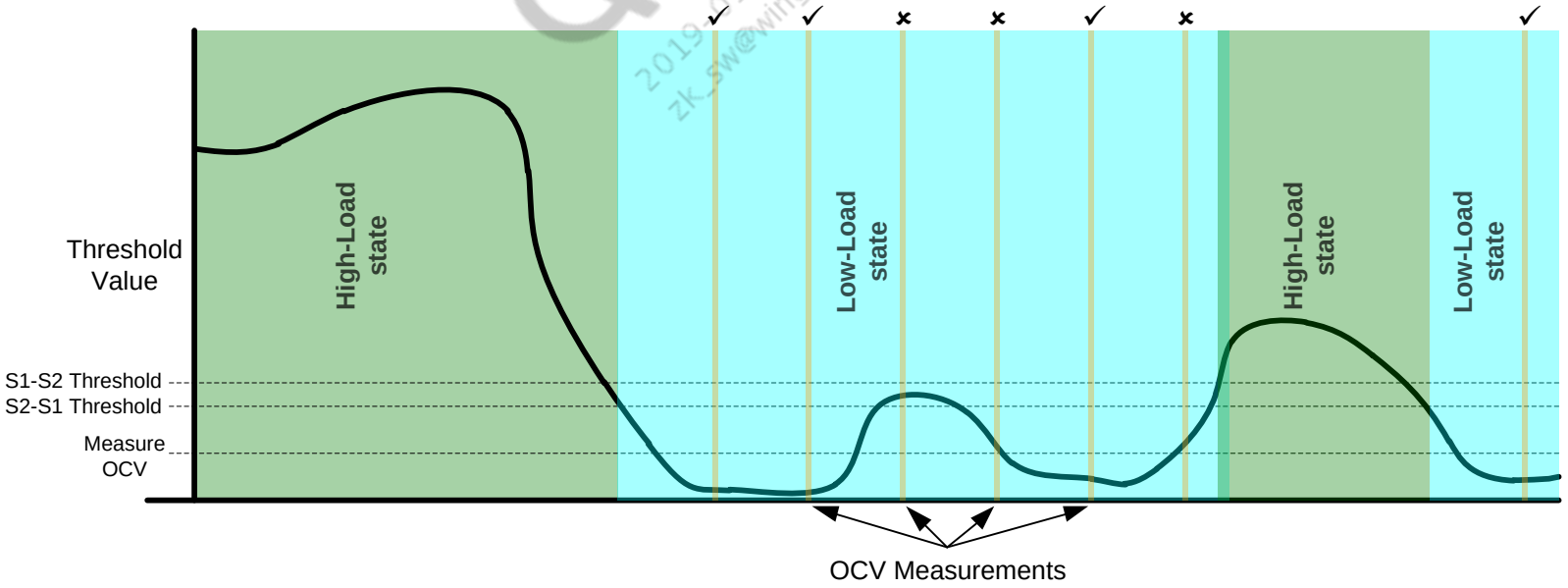
Advanced Hardware Topics – S3 State Timing

- OCV state (S3) timing

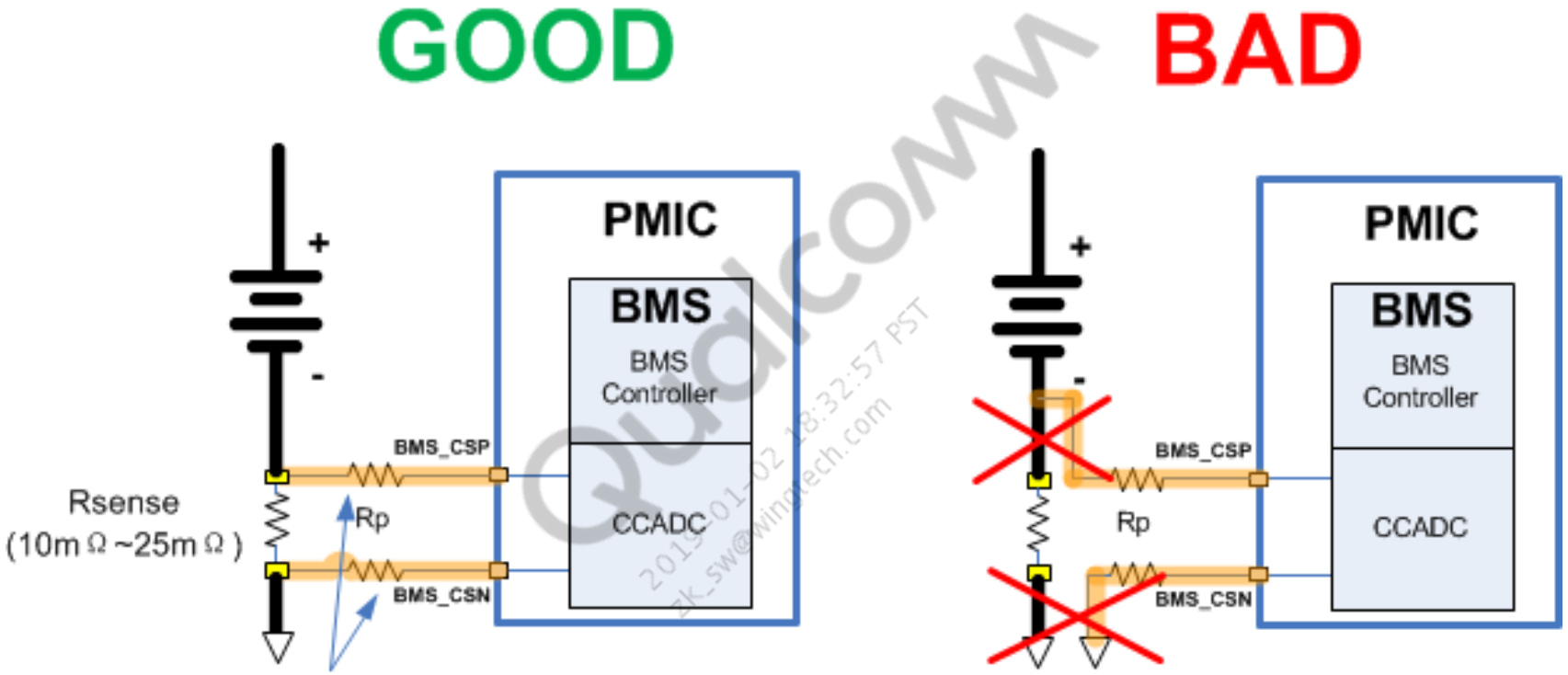


Advanced Hardware Topics – BMS Controller

- The BMS control algorithm relies only on counter outputs for state changes.
  - Simplification of FSM adds robustness – unexpected inputs are impossible.
- The algorithm uses two main states and one auxiliary state.
  - Each state has one V_{sense} threshold level.
  - Main states: S2 – high-load (active); S1 – low-load (standby)
  - Auxiliary state: S3 – OCV measurement
    - VADC is only used during the auxiliary state.



Advanced Hardware Topics – BMS PCB Layout Guidelines



BMS Trace Routing Rules:

1. BMS_CSP must connect to negative terminal of battery. Don't change polarity!
2. Route traces directly to Rsense pad without sharing any power trace!
3. Keep trace Resistance (Rp) less than 1Ω.
4. Route traces close together, with common length and common resistance.
5. Maintain similar thermal coupling.
6. Keep away from high power traces and devices.
7. Place the sense resistor in the most optimal location between the battery and the PMIC without adding unnecessary trace resistances to battery leads, yet minimizing the length of the sense traces.

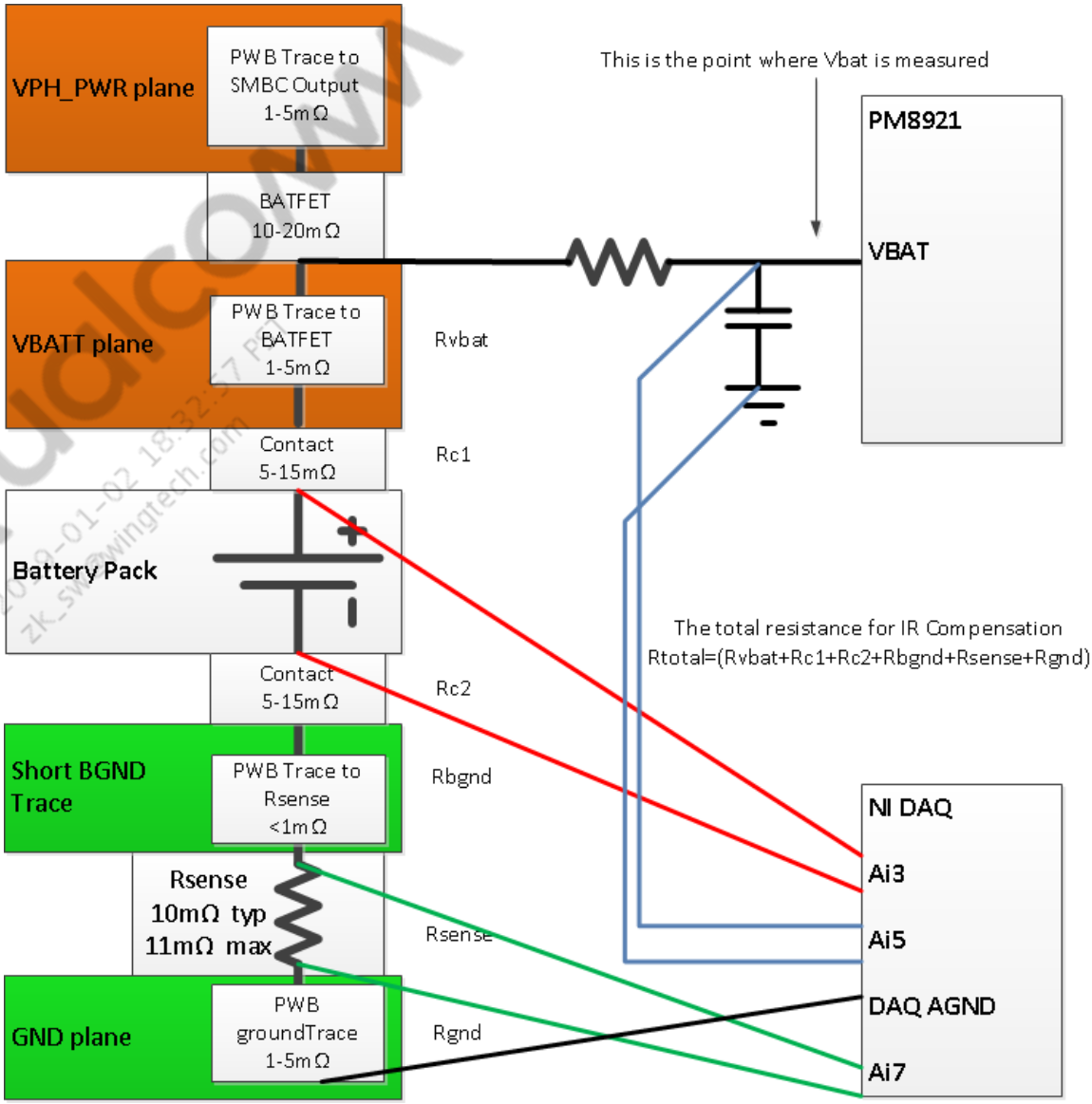
Key Test/Use Cases – Standalone Bench Test

- The following table summarizes the measured SoC accuracy on a standalone PMIC device.
- The measured SoC accuracy is < 3%.

Ambient temp (°C)	Static load condition			GSM load condition		
	Load (Battery C rate)	CC accuracy (%)	SOC accuracy (%)	Load (A)	CC accuracy (%)	SOC accuracy (%)
25	0.5 (C/3)	-2.32	2.16	1.5	-2.34	2.14
25	1.125 (0.75C)	-2.3	2.13	2	-2.26	2.17
25	1.5 (1C)	-2.34	1.98	2.5	-2.3	1.56
-10	0.5 (C/3)	-1.95	1.06	1.5	-2.02	0.54
-10	1.125 (0.75C)	-1.97	0.53	2	-1.96	-0.35
-10	1.5 (1C)	-1.85	0.19	2.5	-1.44	-0.11
40	0.5 (C/3)	-2.4	2.27	1.5	-2.4	2.23
40	1.125 (0.75C)	-2.58	2.46	2	-2.42	2.27
40	1.5 (1C)	-2.4	2.23	2.5	-2.34	1.65

Key Test/Use Cases – Test Configuration

- The National Instruments NI-USB-6218 was used as the SoC reference.
- The DAQ was connected across Rsense to collect simultaneous measurements.
- All SoC error results are based on BMS SoC relative to the DAQ SoC values.
- All DAQ measurements are differential with the AGND referenced to system ground, as shown.



Key Test/Use Cases – Systems-Level Test Data

- Several primary use-case results are provided.
- SOC error is within 1%.

Use case	Typical SOC error (%)	Platform	Test conditions	Test duration
GSM burst discharge	0.8	MTP	1.5 A peak current, continuous GSM call, no OCV	6.1 hours
Mixed-mode discharge/charge	0.4	Fluid	Discharged: Neocore, GSM call, WiFi/BT enabled, Charged: Via USB with GSM registered, Wifi/BT enabled	9.7 hours
Wall-charge/sleep/ multiple GSM calls	-0.9	Fluid	Charge via wall charger, sleep, wake on GSM call, repeated GSM calls. Charger automatically recharges on Vbat dip.	13.9 hours
48-hour standby	-0.1	Fluid	Fluid, standby operation	48.0 hours

Use case:
Neocore, GSM call

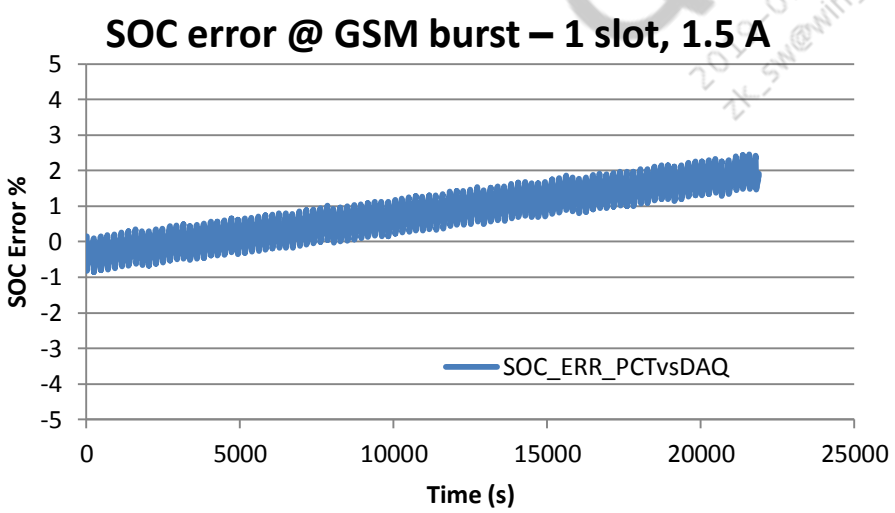
Platform:
MTP

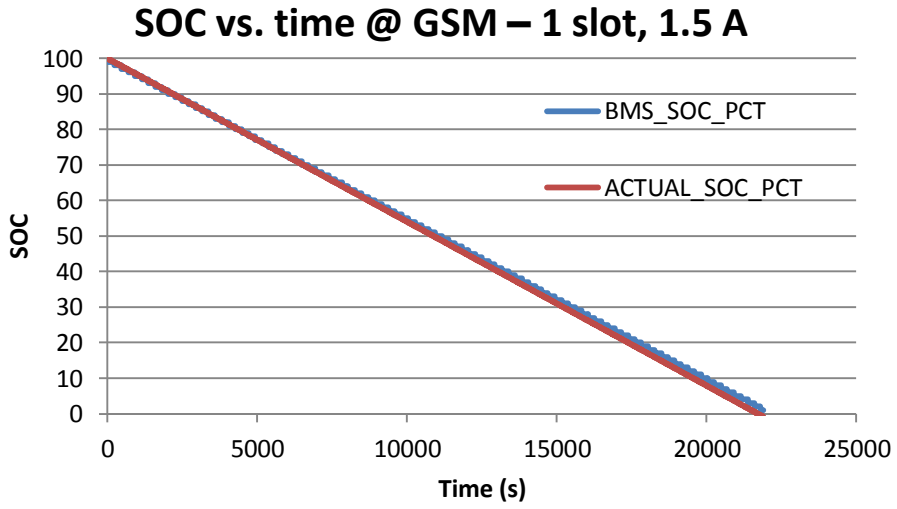
SOC start %	SW est. OCV stop (V)	Measured OCV stop (V)	OCV stop error (mV)	Starting SOC error (%)	Ending SOC error (%)	Avg. SOC error (%)	Test duration
99	3.593	3.572	-28	0.01	0.49	-0.73	2.1 hours
77	3.587	3.528	-72	0.45	0.10	-0.01	1.0 hours
48	3.571	3.540	-60	7.02	-0.29	3.33	0.8 hours
32	3.567	3.591	-9	8.89	-0.24	4.24	0.4 hours

Key Test/Use Cases – System-level Test Data

GSM burst-discharge test description

- Modem Test Platform (MTP) running Linux Android
- Real battery – 1500 mAh
- GSM burst, 1/8 duty cycle, peak current of 1.5 A
- Temperatures tested: 25 C
- Coulomb counting accuracy, no OCV updates

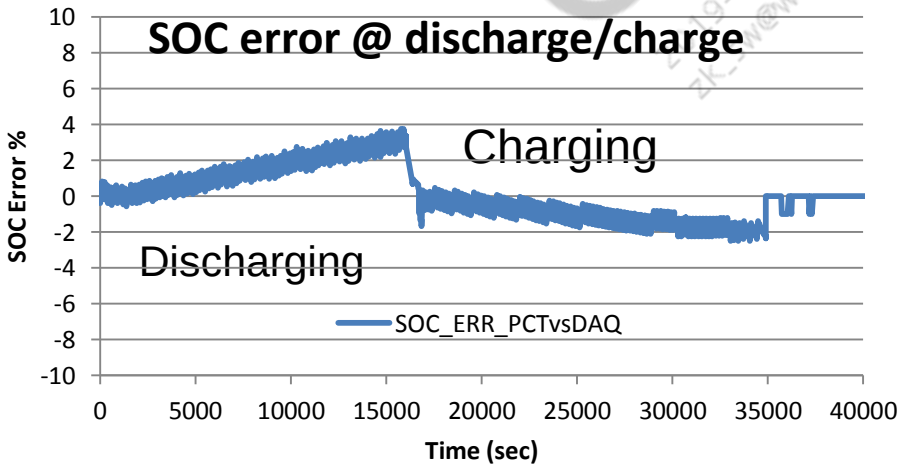




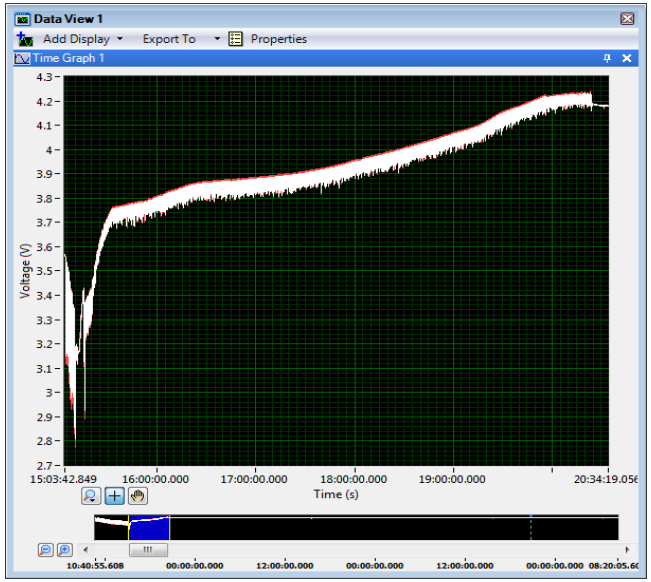
Key Test/Use Cases – System-level Test Data *(cont.)*

Mixed-mode discharge

- FLUID running Linux Android.
- Real battery – 1500 mAh.
- Discharge running Neocore demo, LB GSM call at max Pout.
- Charge via PC USB while registered to GSM, WiFi/BT enabled.



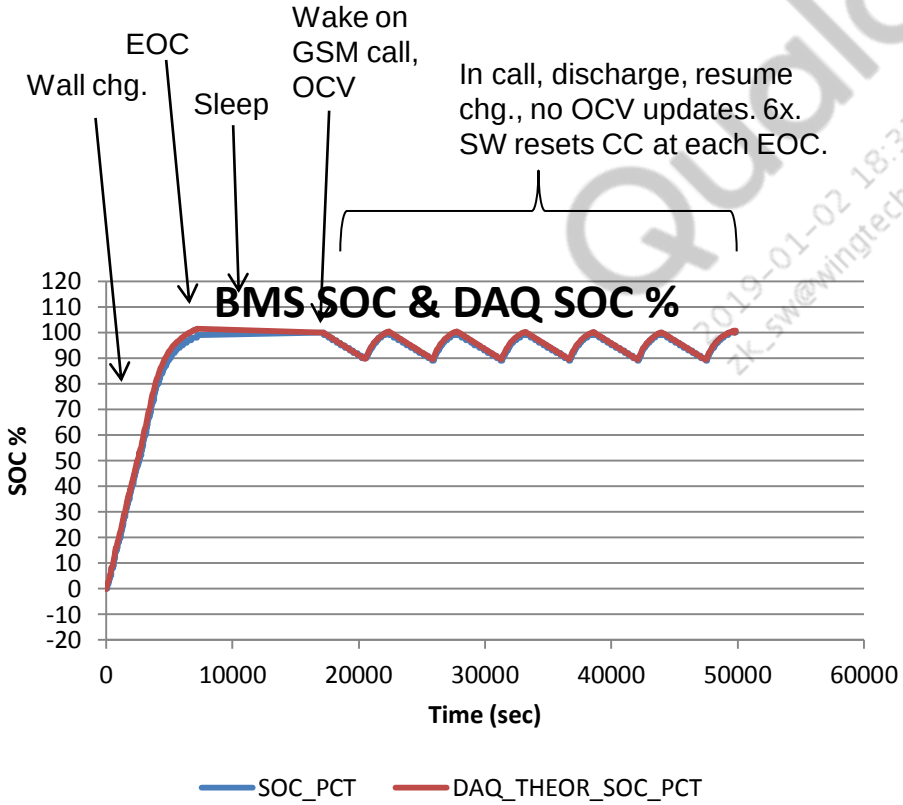
Battery voltage during charge

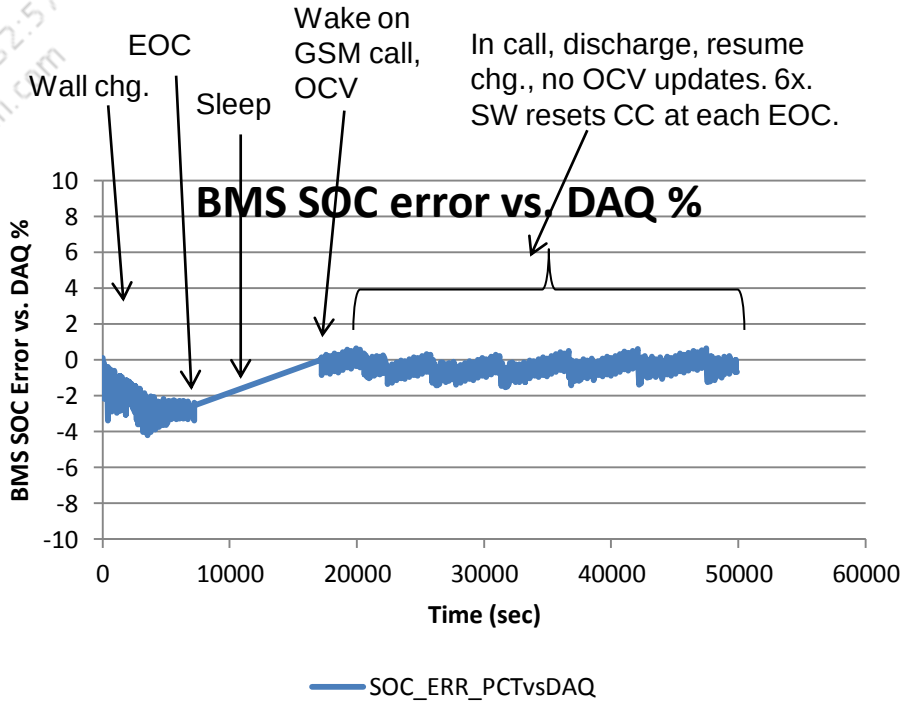


Key Test/Use Cases – System-level Test Data *(cont.)*

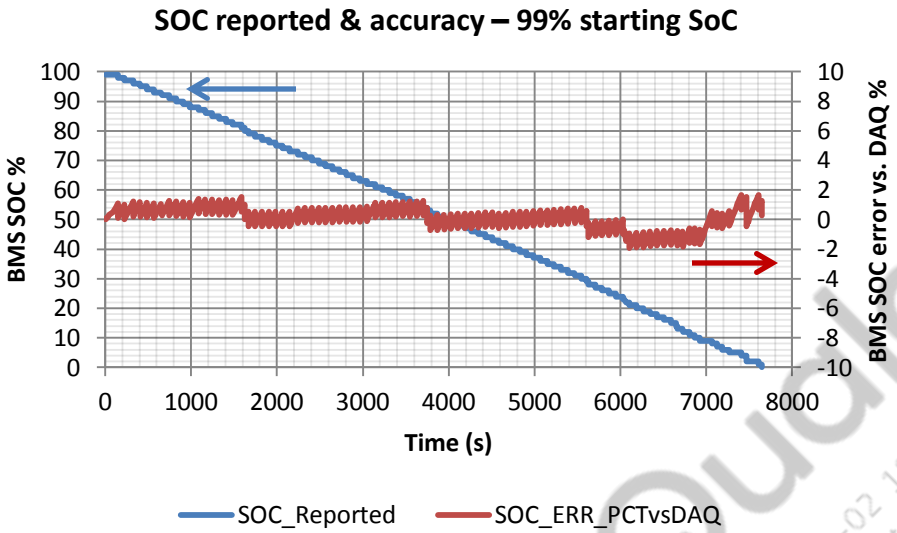
Wall charge / sleep / GSM calls

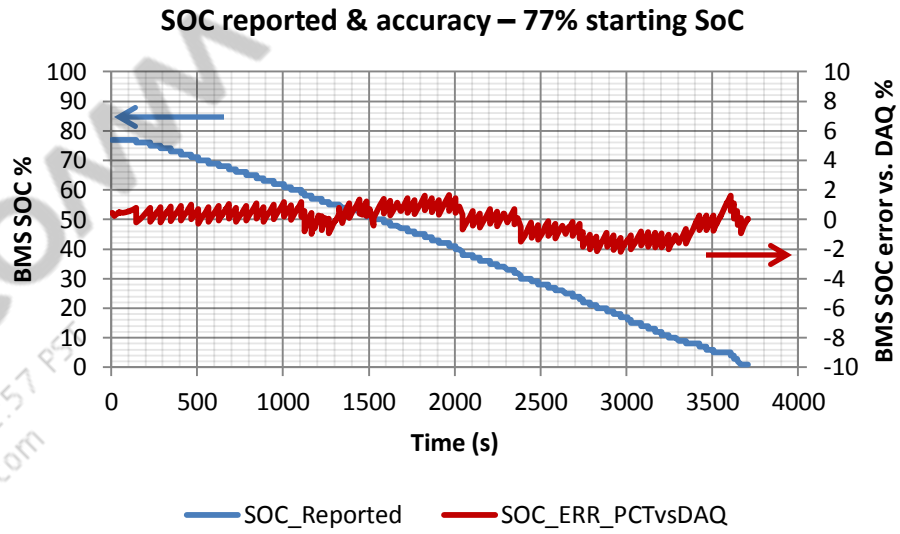
- FLUID running Linux Android, using 1500 mAh battery.
- Discharge running Neocore demo, LB GSM call at max Pout.
- Charge to 100% via wall charger while registered to GSM, WiFi/BT enabled.
- Sleep, wake with GSM call, and continue repeated GSM calls.

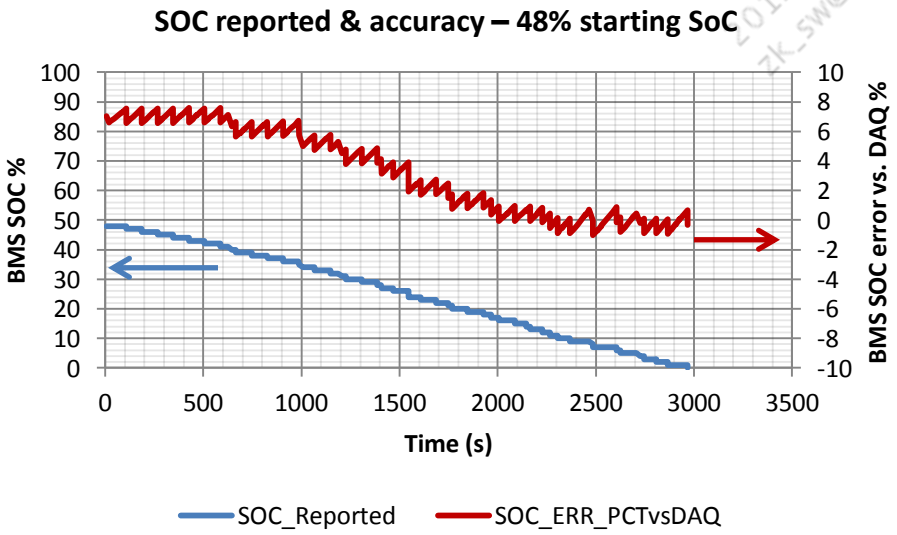


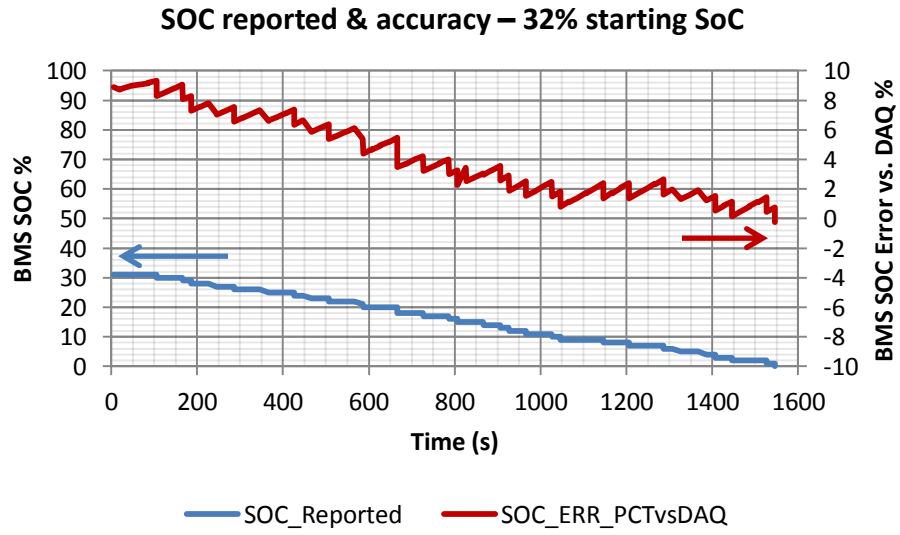


Key Test/Use Cases – System-level Test Data *(cont.)*



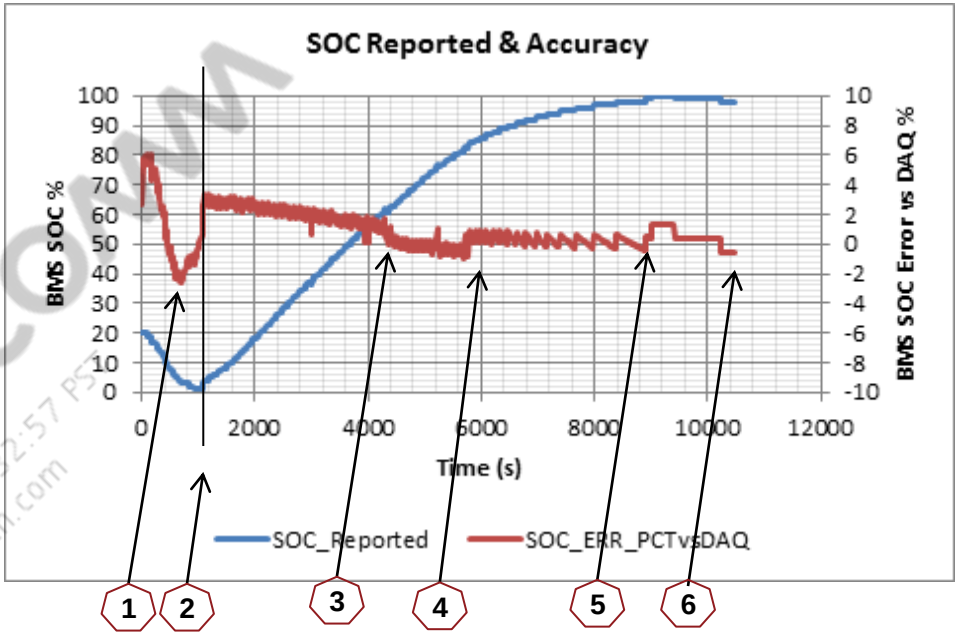






Key Test/Use Cases – System-level Test Data *(cont.)*

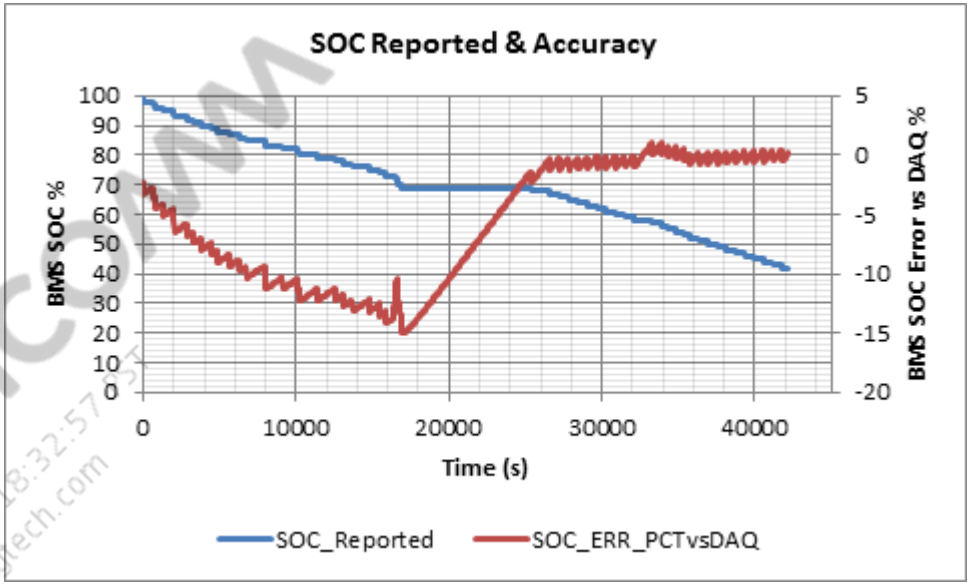
- Typical use case on FLUID platform
- Data includes SOC error correction, IR compensation, and latest dynamic UUC correction.
- Test scenario:
  - Keypad PON. PON OCV **error = -3 mV** with SOC at 20%.
  - WLAN + BT enabled.
  - GSM call + NeoCore demo running until 1% SOC.
  - Wall charger inserted.
  - Stop NeoCore + GSM call.
  - Continue wall Charge until EOC, 100% SOC.
  - Standby, wall is supplying system, collect OCV updates.
  - Unplug wall charger, keypad power off. Settle battery for 20 minutes to compare with SW OCV estimate.
  - OCV_Est error vs. DAQ settled VBAT = **2 mV**.
- SOC error is referred to settled VBAT at POFF.

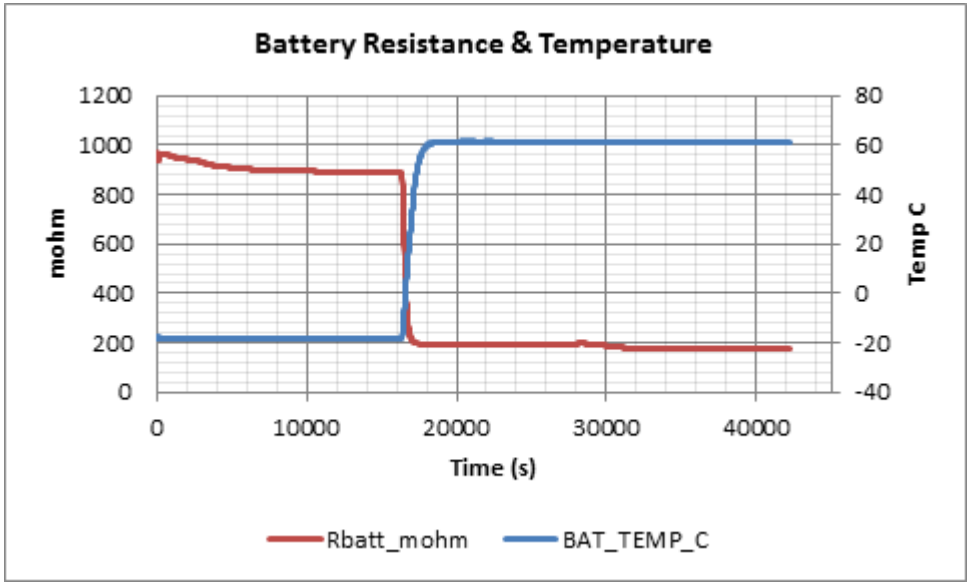


1. SOC error correction degrades accuracy quickly due to battery model lacking R3/C3. During last 5%, the correction algorithm corrects itself back towards 0% error.
2. Wall charger inserted at 1% while in call and NeoCore.
3. Stopped call and NeoCore, display on.
4. Charging with minimal load, display off. Sawtooth is VDD_MAX adjustment algorithm.
5. EOC, begin OCV updates.
6. Unplug charger, keypad power off.

Key Test/Use Cases – System-level Test Data *(cont.)*

- Customer use case is one that requires the phone to:
  1) Be in low-load sleep or standby for 4 hours (14400 s) at -20 C (~45 mA average Ibat here).
  2) Start high-load call or equivalent.
  3) Temperature ramp to 60 C.
  4) Be in a low-load sleep or standby for 4 hours at +60 C (~95 mA average Ibat here)
  5) Start high-load call or equivalent.
- Conclusions
  - SOC error correction degrades error too quickly, due to the current Rbat model. Long-term solution for Rbat model should fix this (discussed in later slide).
  - SOC accuracy at beginning and end points are fine.
  - **>4 hours at -20 C with SOC >70%.**



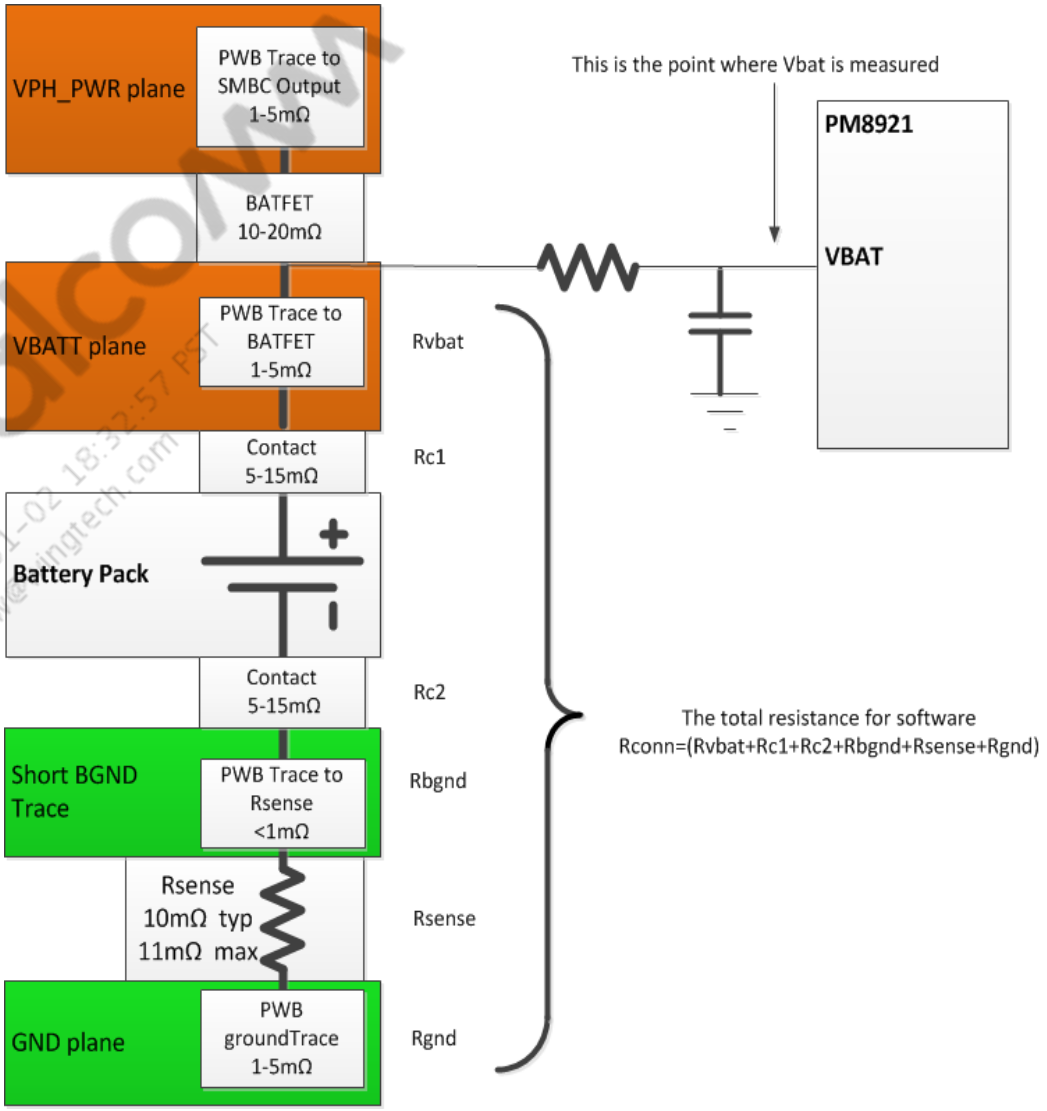


Advanced BMS Algorithms – Dynamic SoC Error Correction

- A SoC error-correction algorithm was developed for improving the SOC error at the end of battery life, to allow the phone to shut down at an appropriate point and keep the system out of UVLO.
- This algorithm uses simultaneous battery voltage and current measurements to compensate for SOC error.
- The following slides describe the current algorithm in detail.

Advanced BMS Algorithms – Dynamic SoC Error Correction *(cont.)*

- The impedance of the actual PCB was also not accounted for previously in the SoC calculation.
- **Solution:** A variable has been created in the BMS software for compensation (i.e., R_{CONN} in Linux Android).
- An example for how to calculate PCB resistance is shown on the right (FLUID).



Advanced BMS Algorithms – Dynamic SoC Error Correction *(cont.)*

- When the SoC is requested by software to be reported, the BMS takes a simultaneous Vbat and Vsense measurement in order to derive an instantaneous estimated OCV measurement (OCVest)

- **Detailed steps 1 to 5:**
  1. Calculate the SOC, as is currently done.
  2. Put BMS into override mode, with the simultaneous V_{bat} bit set.
  3. Read the values for V_{bat} and V_{sense}.
  4. Put the BMS back into autonomous mode.
  5. Apply calibration to both values.

Advanced BMS Algorithms – Dynamic SoC Error Correction *(cont.)*

- Before OCVest can be calculated, the proper effective battery resistance (Rbatt_eff) must also be calculated.
- Also, the resistance of the board must be determined (R_{CONN}).
- Additional compensation is incorporated (ΔR) to address Rbatt_eff.

- **Detailed steps 6 to 8:**

6. Calculate Rbat_eff = Rbat + R_{CONN} + ΔR , where Rbat is the lookup value, R_{CONN} is the measured board resistance, and ΔR is used to compensate for additional resistance and capacitance (R3 and C3), where:

   If 20>SOCrbatt>10, then ΔR=60 mΩ *(20-SOC)/10,

   elseif SOC<10 ΔR=60mΩ,

   else ΔR=0
7. Convert V_{sense} to current value (I_{bat}).
8. Create $OCV_{est}=V_{bat}+I_{bat}*R_{bat\_eff}$.

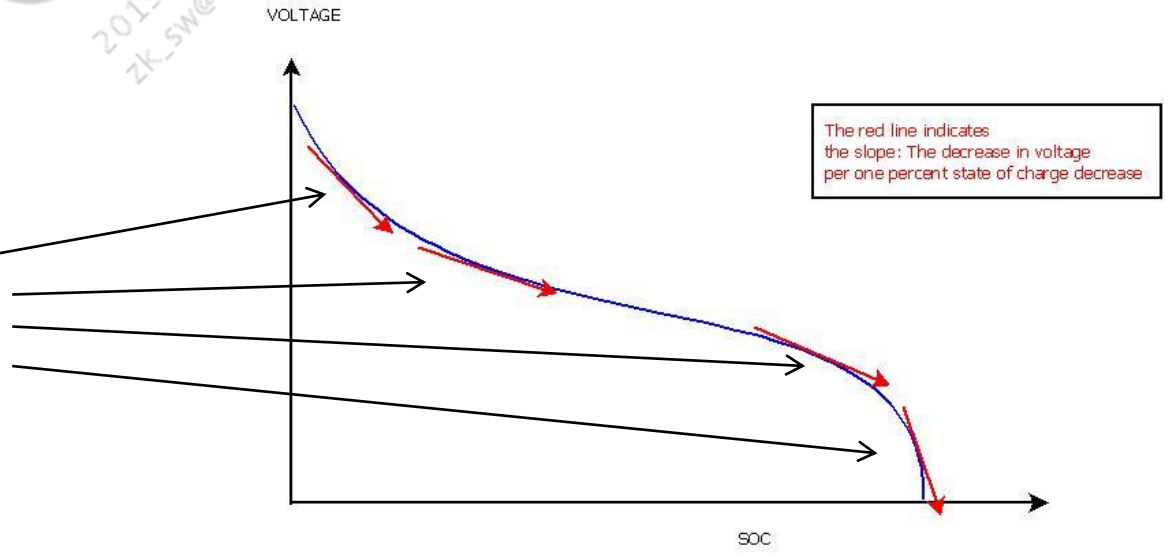
Advanced BMS Algorithms – Dynamic SoC Error Correction *(cont.)*

- An estimated SOC value (SOCest) is then calculated from OCVest.
- From there, an IIR filter is used to allow SOC to approach SOCest.
  - The approach is slower at higher SOC values.
  - The approach is faster at lower SOC values.
- The N value below dictates how many steps it takes for SOC to approach SOCest.

- **Detailed steps 9 to 11:**

9. Map OCV_{est} to SOC_{est}.
10. $N = \min(200, \max(1 , SOC + SOC_{est} + last\_SOC_{est}))$.
11. $\Delta OCV = M*(SOC_{est}-SOC)/(N)$, where M = slope of battery profile curve near OCV point (details shown below).

M (the slope) will be larger at higher or lower SoCs. Around the flatter part of the battery curve, M will be smaller.



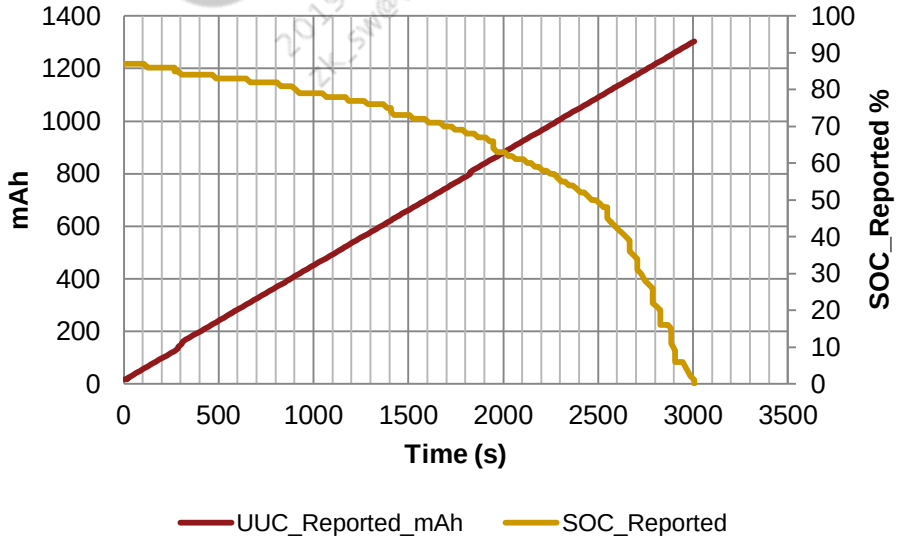
Advanced BMS Algorithms – Dynamic SoC Error Correction *(cont.)*

- A new OCV value is then calculated, based on the addition of the ΔOCV calculated.
  - Finally, a new SoC value is determined based on the updated OCV value.
  - To avoid the potential of premature shutdown, SoC will continue to be reported as 1%, under the scenario where SOC has been calculated to be 0%, but SOCest has not yet reached 0%.

  - **Detailed steps 12 to 14:**
    12. Set OCV = OCV+ ΔOCV.
    13. Calculate SOC using the updated OCV value.
    14. If SOC = 0 and SOCest > 0, then SOC = 1 (avoids premature shutdown).

Advanced BMS Algorithms – Dynamic UUC

- Some carriers require 4-hour operating time at -20 C with low current.
  - Battery resistance is very high when cold.
  - UUC is determined by battery resistance and peak system current.
  - UUC can be so large that the battery reads 0% SoC for fully charged battery.
  - The solution needs to maintain 4-hour operation.
- Software algorithm created to change from fixed UUC to dynamic UUC.
  - UUC is not only based on the peak system current setting (Itest).
  - UUC also refers to the online measurement of the average current draw (Iavg).



Advanced BMS Algorithms – Dynamic UUC *(cont.)*

1. **Calculate a UUC value using Iavg.**
   - ***uuc_uah_iavg = FCC * Lookup(v_failure + Iavg*rbatt)***

2. **Calculate a UUC value using Itest.**
   - ***uuc_uah_itest = FCC * Lookup(v_failure + Itest*rbatt)***

3. **A stepsize value is calculated based on whether the battery is charging or discharging.**
   - ***stepsize = max(1, (SOC_RBATT_CHG - soc_rbatt)) when charging***
   - ***stepsize = max(1, (soc_rbatt - SOC_RBATT_DISCHG)) when discharging***

4. **Calculate the delta_uuc between the Iavg and Itest based UUC.**
   - ***delta_uuc = (min(uuc_uah_itest, uuc_uah_iavg) –last_uuc_reported)/stepsize***

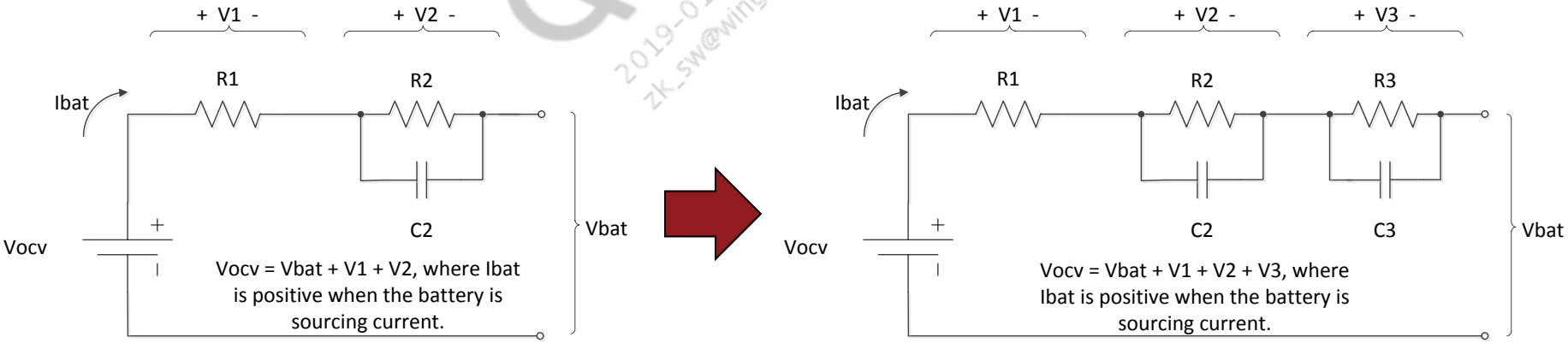
5. **Add delta_uuc to the last_uuc_reported to determine a new uuc_reported**
   - ***uuc_reported = last_uuc_reported + delta_uuc***

6. **Note that if calculating uuc_reported for the first time without having previous UUC knowledge, last_uuc_uah is initialized at ucc_uah_itest if charging, and uuc_uah_iavg if discharging.**

7. **uuc_reported will factor into the reported SoC.**

Ongoing BMS Improvements

- Battery model
  - Improving battery model to include second time constant
    - First time constant is short (30 to 60 seconds).
    - Second time constant is long (10 to 20 minutes).
    - Needed for periods of extended current consumption.
  - Updating battery profiling method to capture individual model parameters.
  - Updating SOC calculation and error correction to use updated model.
  - FCC learning must be verified with updated model and correction algorithm.



Current battery model

Improved battery model

Next-generation Planned BMS Improvements

- The following list of improvements is being added to PM8941.

Improvement/ change	Description	Benefit
Integrated Rsense	Place Rsense within PMIC.	Reduce eBOM and PCB area.
Improve poweron OCV	Improve master band-gap settling time and voltage error.	Reduce poweron OCV error and resulting SoC error.
Configurable OCV interrupt thresholds	BMS can be configured to interrupt at the specified OCV level.	Provides interrupt-driven SoC.
Coulomb counter to have selectable reset and configurable limits for OCV updates	Add selection capability to reset Coulomb counter at each OCV, or by software only. Add selection to control battery voltage range in which OCV updates are conducted.	Provides ability to avoid OCV updates in the flat portion of the battery curve. Provides ability to maintain Coulomb count during sleep/standby.
Shadow Coulomb counter	A separate dedicated Coulomb counter that is only reset by software.	Provides ability to use combination of OCV and Coulomb count results when exiting sleep/standby.

Questions?

https://support.cdmatech.com

REDEFINING MOBILITY