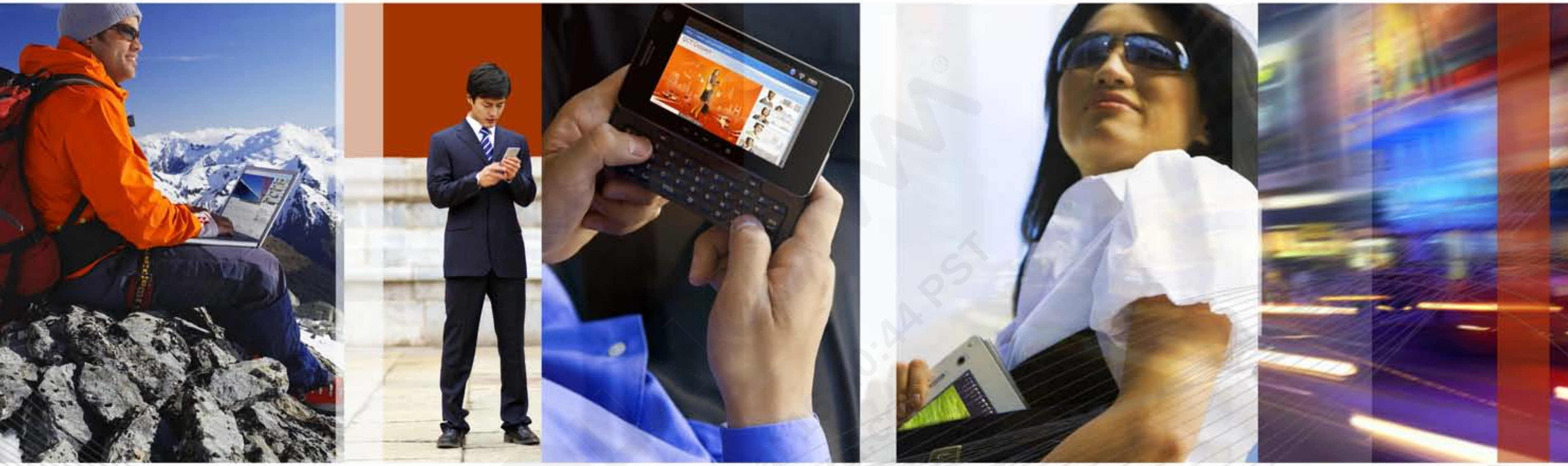


QUALCOMM CDMA Technologies

REDEFINING MOBILITY

# PM8921™ Linux PMIC Interfaces – Charger and BMS

80-N8278-1 A

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

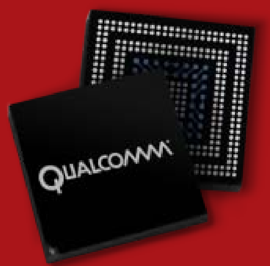
# Revision History

| Version | Date | Description |
|---|---|---|
| A | Nov 2011 | Initial release |



# Contents

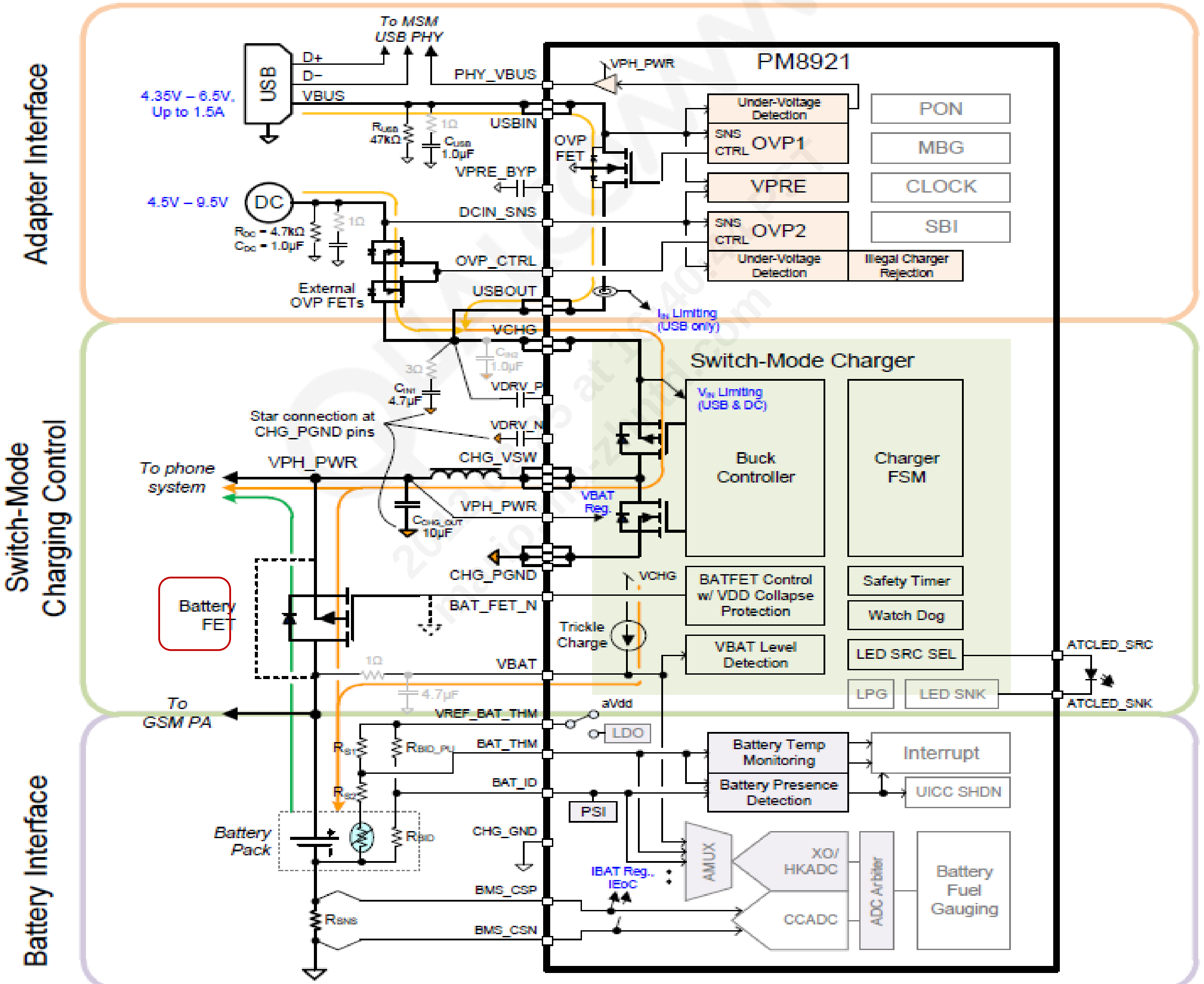


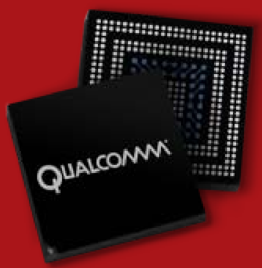

# PM8921™ Power Input Subsystem

# PM8921 Power Input Subsystem

- PM8921 power input system consists of three components
- Adaptor interface
  - USBIN
  - DCIN
- Switch control charging control
  - Trickle charging
  - Fast charging
- Battery interface
  - BAT_ID
  - BAT_THM

# PM8921 Power Input Subsystem Diagram



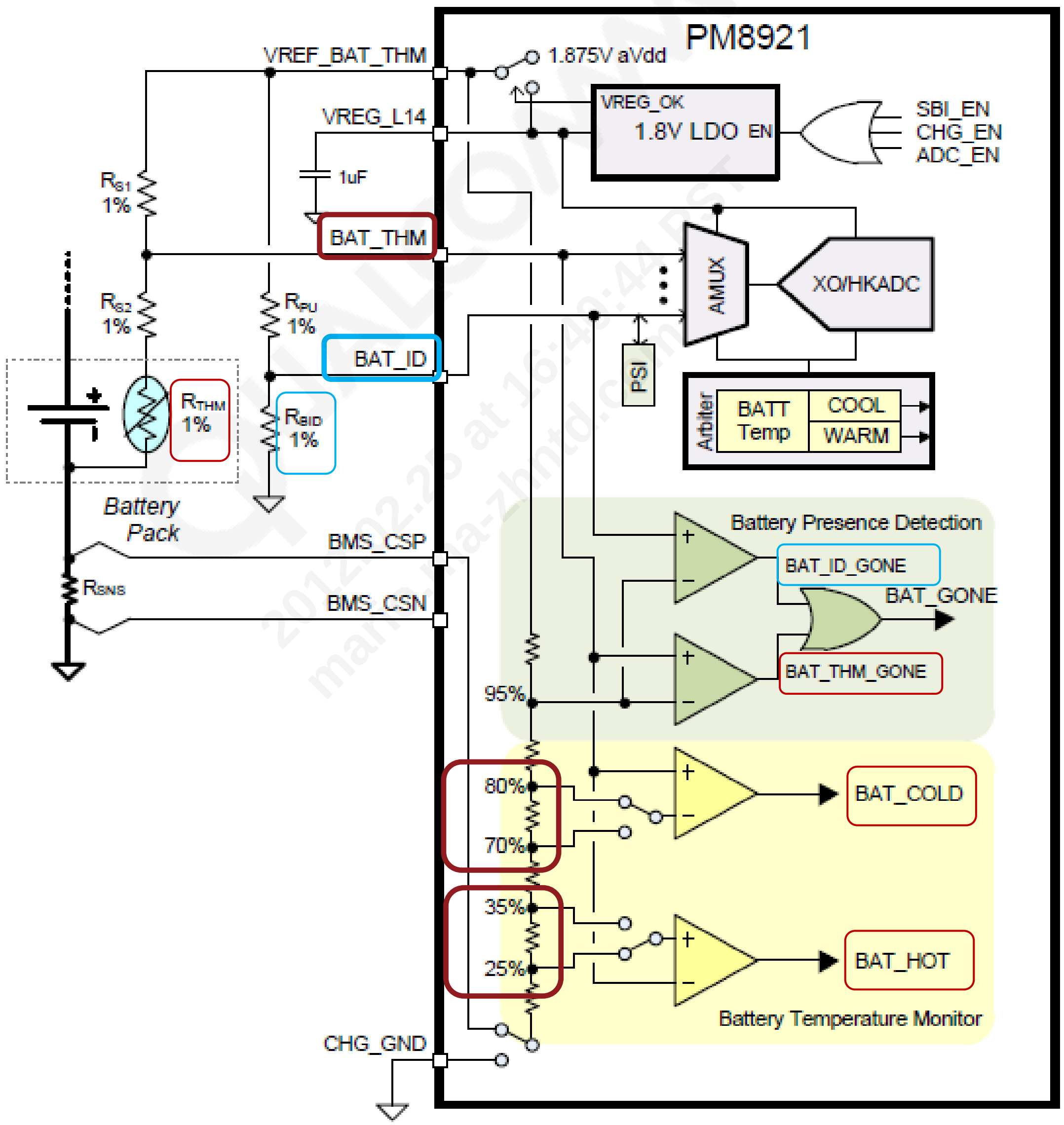



# PM8921 SMBC Charger

# PM8921 Battery Presence Detection – BPD

- The battery has to be present before charging starts.
- PM8921 provides two ways to identify the battery presence:
  - BAT_ID
  - BAT_THEM
- Battery presence is detected by sensing the presence of battery thermistor or ID resistor.
- Two dedicated BPD comparators monitor BAT_THM and BAT_ID voltage levels. Battery is considered as gone if either one is above the 95% threshold.
- If BAT_ID is not used, then the unused pin has to be connected to GND.
- Some customers are unable to start charging since they use BAT_THM to detect the battery presence and forget to ground BAT_ID.

# PM8921 Battery Presence Detection – BPD (cont.)

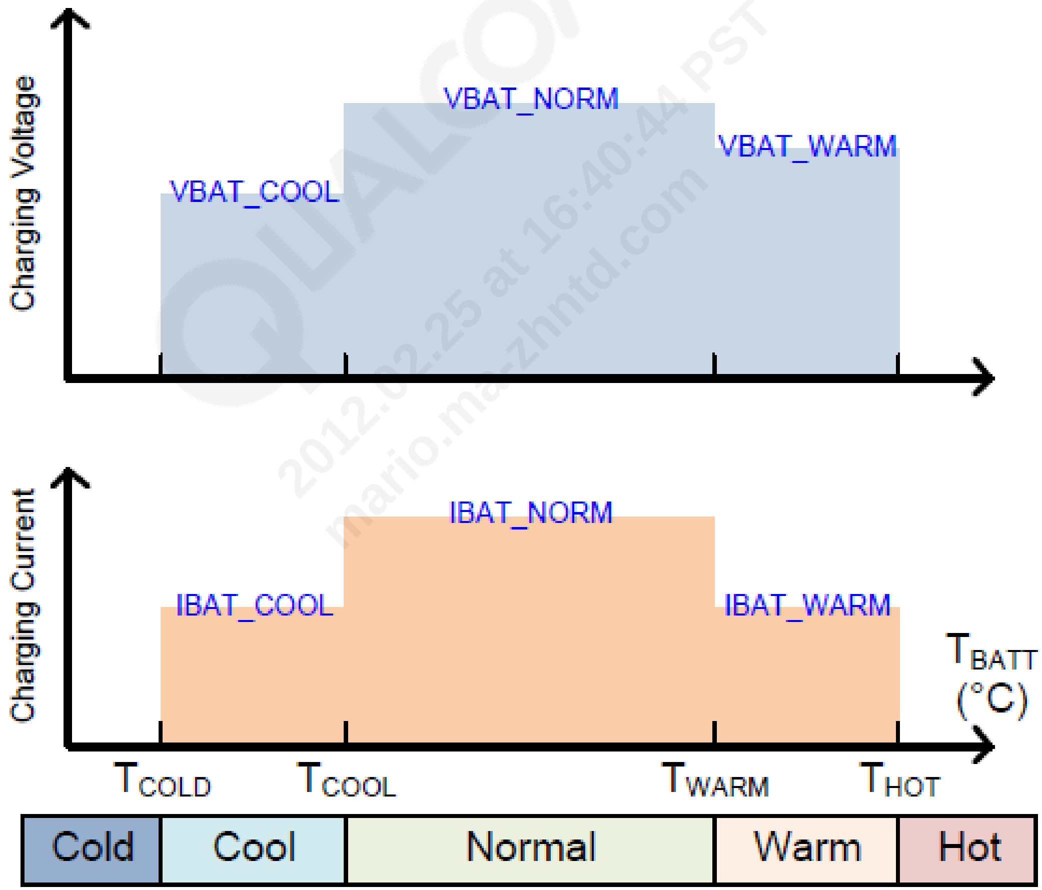


# Battery Temperature Monitoring

- To prevent permanent damage of the battery, battery charging will be stopped if the battery temperature is out of range
- PM8921 BTM is used to monitor the battery temperature
- No BTM during the first hardware-controlled ATC
- The SMBC BTM is disabled before software configures it
  - Enable BTM − pm_chg_masked_write(chip, CHG_CNTRL_2, CHG_BAT_TEMP_DIS_BIT, 0) in m8921_chg_hw_init()
- For customers using external fuel gauge
  - Disable BTM − pm_chg_masked_write(chip, CHG_CNTRL_2, CHG_BAT_TEMP_DIS_BIT, 1) in m8921_chg_hw_init()
- BTM will be in later ATCs if VCOIN > 2 V

# Battery Temperature Monitoring (cont.)

- Temperature Thresholds (JEITA compliance)



# Battery Temperature Monitoring (cont.)

- JEITA compliance – (-10ºC ~ 60ºC)
  - CHG_BATT_TEMP_THR_COLD = 80%
  - CHG_BATT_TEMP_THR_HOT = 20%

```
static struct pm8921_charger_platform_data pm8921_chg_pdata
  __devinitdata = {
        .cold_thr=1; // 80%
      .hot_thr=0;  // 20%
    };
```

- Non-JEITA compliance – (0ºC ~ 45ºC)
  - CHG_BATT_TEMP_THR_COLD = 70%
  - CHG_BATT_TEMP_THR_HOT = 35%

```
static struct pm8921_charger_platform_data pm8921_chg_pdata
  __devinitdata = {
        .cold_thr=0;  // 70%
      .hot_thr=1;   // 35%
    };
```

# Battery Temperature Monitoring (cont.)

- Configuration of cool and warm thresholds (JEITA compliance only)

```
static struct
pm8921_charger_platform_data pm8921_chg_pdata __devinitdata =
{
  .cool_temp              = 10,    // 10 degree celsius
  .warm_temp              = 40,    // 40 degree celsius
  .cool_bat_chg_current   = 350,    // 350 mA  (max value = 2A)
  .warm_bat_chg_current   = 350,    // 350 mA
  .temp_check_period      = 1      // 1 second (max value = 16
 seconds)
};
```

# Selection of Thermistor Pull-Up Resistors ($R_{S1}$ and $R_{S2}$)

1. Find the battery thermistor parameters – Room temperature, resistance ($R_0$), and temperature coefficient (B)
2. Determine the allowable battery charging temperature range, e.g., 0°C ($T_{COLD}$) to 40°C ($T_{HOT}$)
3. Calculate the thermistor resistance at cold and hot
   – $R_{COLD} = R_0 \cdot \exp(B \cdot (1/T_{COLD} - 1/T_0))$
   – $R_{HOT} = R_0 \cdot \exp(B \cdot (1/T_{HOT} - 1/T_0))$
4. Select the BTM comparators thresholds
   – For traditional battery charging temp window, such as 0°C to 40/45°C, the cold and hot thresholds should be set to 70% and 35% respectively.
   – For JEITA-compliant extended battery charging temp window, such as -10°C to 60°C, select 80% and 20% as the cold and hot threshold.

# Selection of Thermistor Pull-Up Resistors ($R_{S1}$ and $R_{S2}$) (cont.)

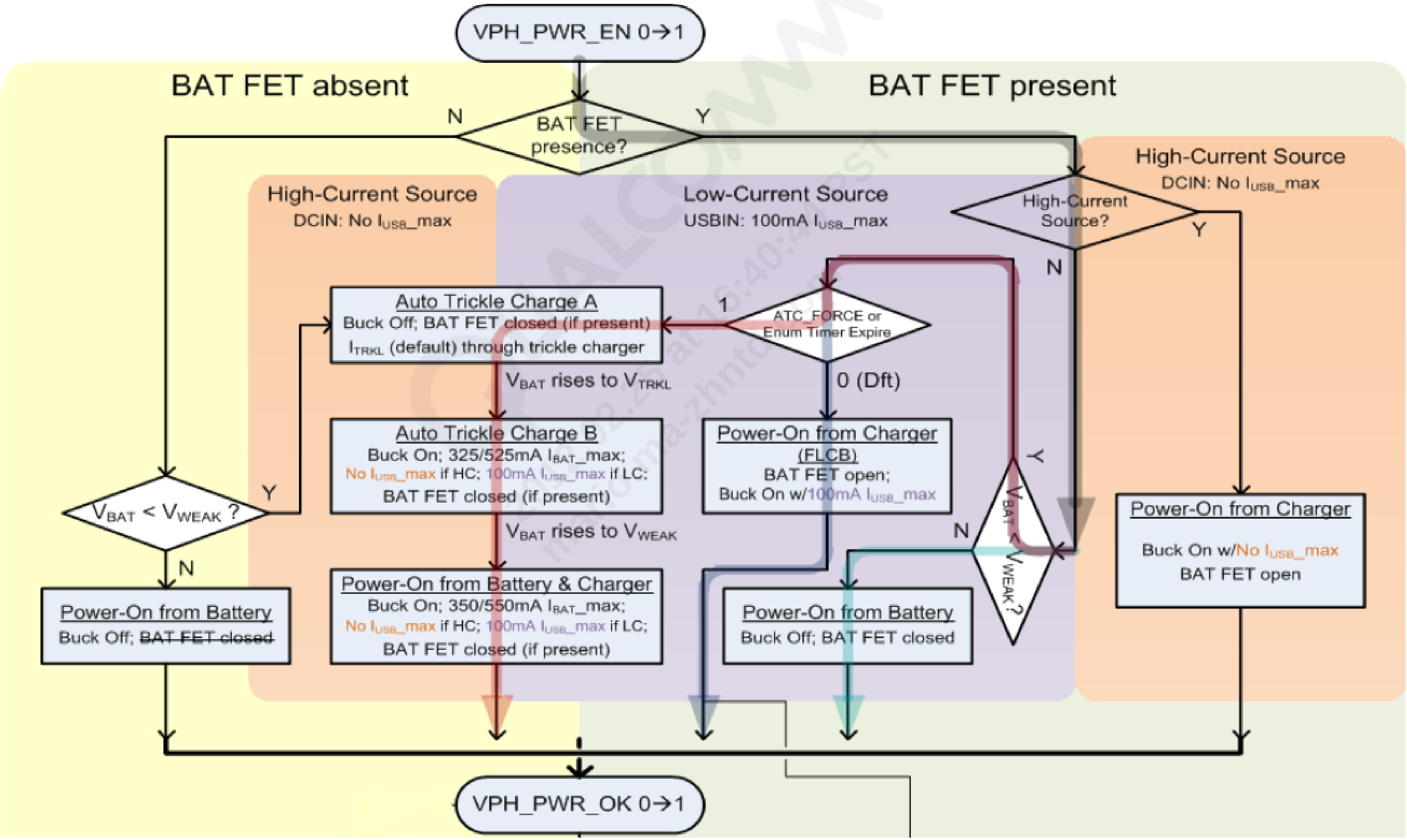
- Another root cause that the customer cannot start the charging
  - If the customer selects the wrong resistor, the charger may not start because the PMIC chip will incorrectly regard the temperature as too cold or too hot.

| Battery charging temperature window | BTM comparators thresholds | $R_{S1}$ and $R_{S2}$ calculations |
|---|---|---|
| 0°C to 40/45°C | 70%/35% | $R_{S1} = \frac{39\,(R_{COLD}-R_{HOT})}{70}$<br>$R_{S2} = \frac{3\,R_{COLD}-13\,R_{HOT})}{70}$ |
| -10°C to 60°C | 80%/20% | $R_{S1} = \frac{4\,(R_{COLD}-R_{HOT})}{15}$<br>$R_{S2} = \frac{R_{COLD}-16\,R_{HOT})}{15}$ |

# PM8921 Charging Process

- Once the battery presence is detected, the charging process starts as follows:
  - If battery voltage is below the weak threshold, Vweak, then trickle charging is performed to prevent the damage of the battery.
  - Otherwise fast charging starts.

# PM8921 Trickle Charging Flowchart

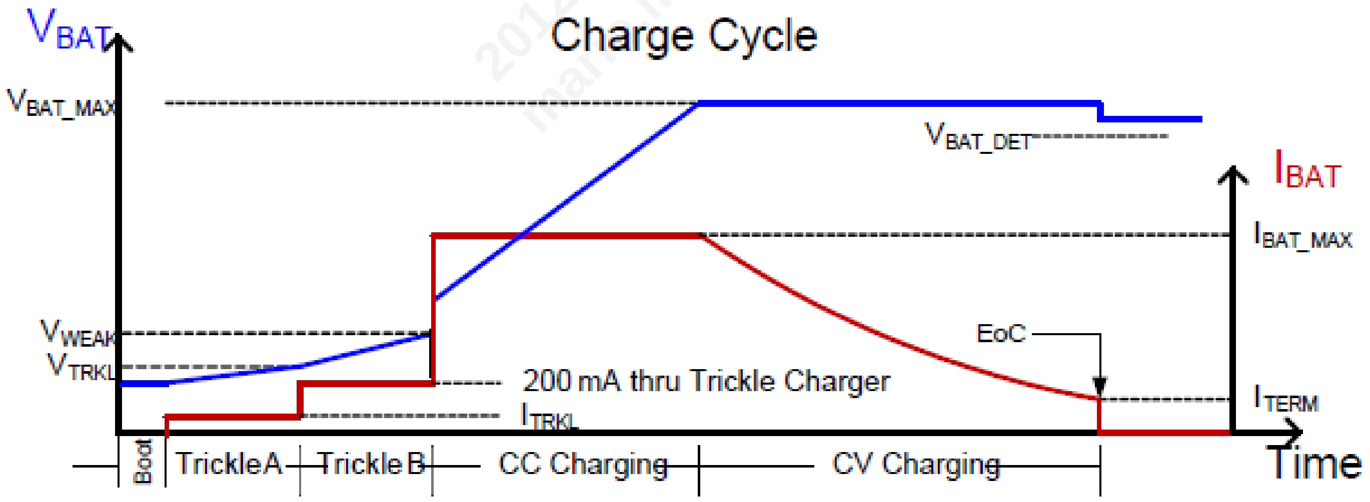


# PM8921 – Charger Driver

- When the phone boots into the Linux kernel, the software driver will do two things
  - Configure the charger parameters
    – vddmax
    – ibatmax
    – Iterm
    – Vbatdet …
  - Start the hardware-controlled fast charging
    – Call pm8921_chg_hw_init() in pm8921-charger.c

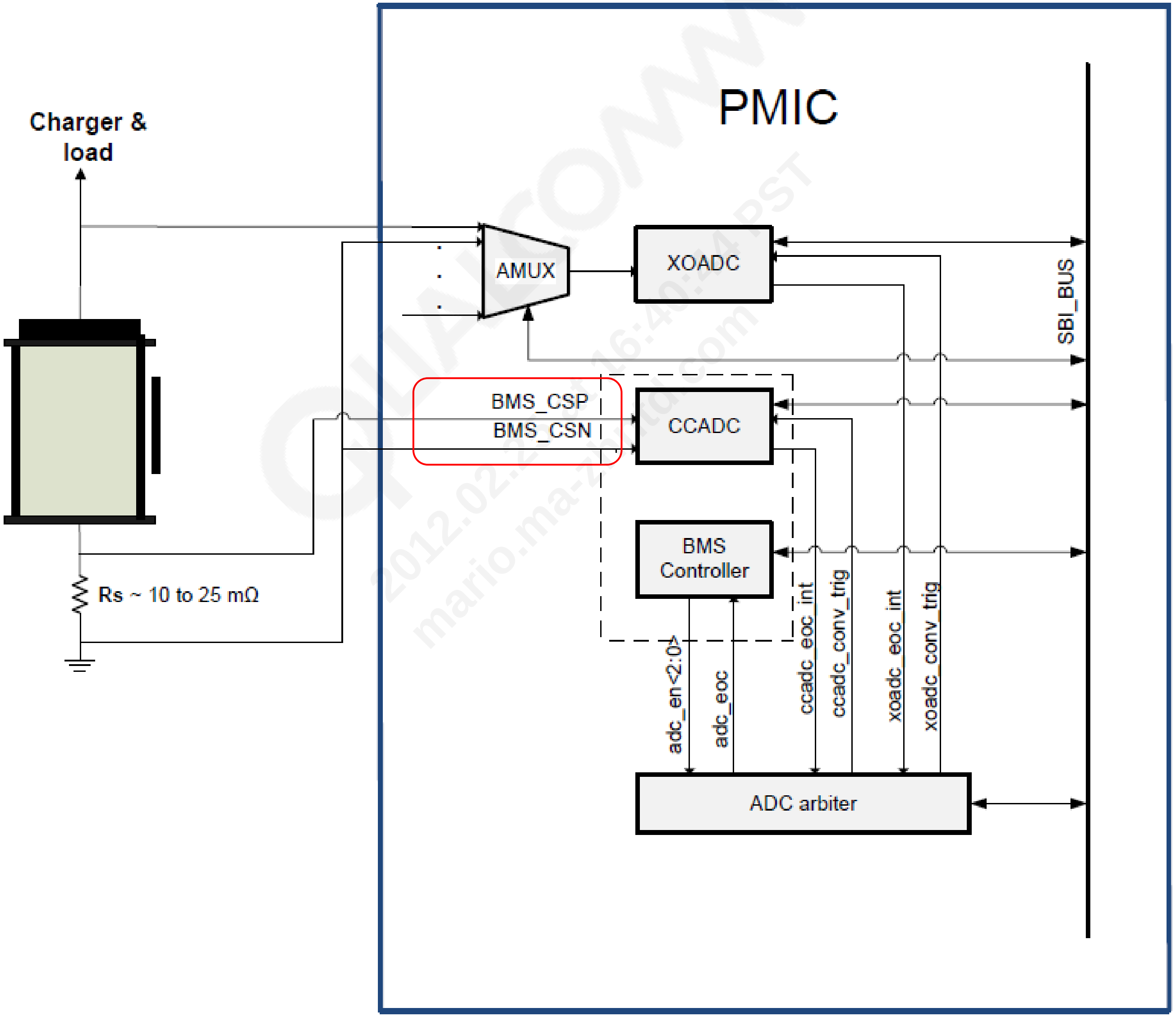
# PM8921 Hardware-Controlled Fast Charging

- PM8921 adopts hardware-controlled fast charging.
- Fast charging consists of two parts:
  - Constant current charging
  - Constant voltage charging
- When the $V_{BAT}$ reaches the threshold $V_{BAT\_DET}$, the charging state will change from constant current charging to constant voltage charging.
- Charging will stop when the $I_{BAT}$ is less than the threshold $I_{TERM}$.




# PM8921 Fuel Gauge – BMS

# PM8921 Battery Monitoring System

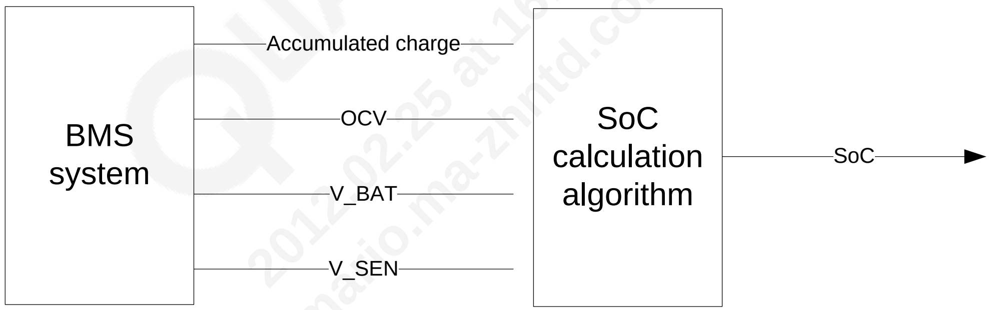


# BMS Components

- Coulomb Counter Counting Accumulated Charge (CCADC)
  - Analog-to-Digital Converter (ADC)
  - Produce digitized Vsense
- Crystal Oscillator ADC (XOADC)
  - Produces digitized Vbatt
- BMS controller
  - Controls the turn-on and turn-off of the analog front-end
  - Determines what data (Vsense, Vbatt) is necessary at what time for accurate State-of-Charge (SoC)
  - SoC approximation software is located on a processor

# PM8921 Battery Monitoring System

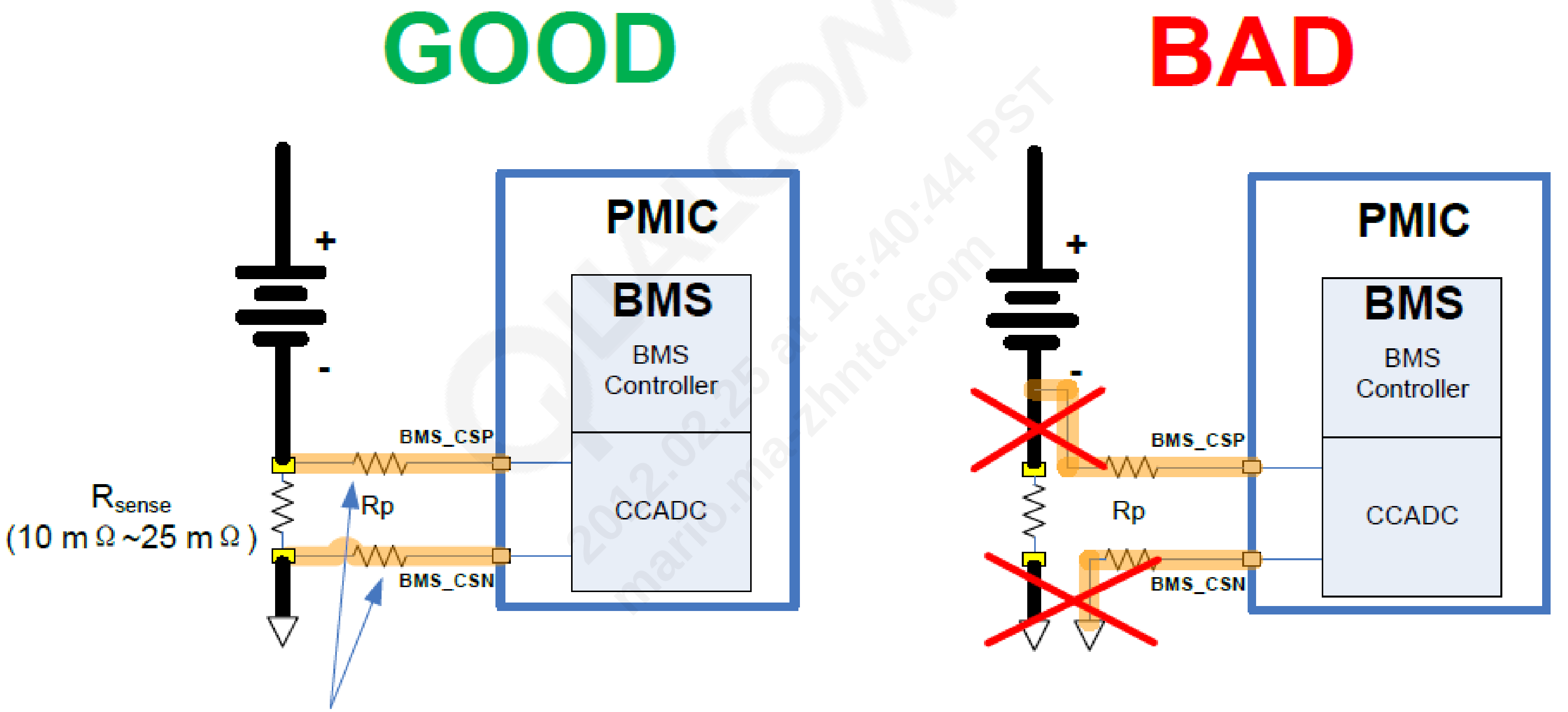
- The purpose of BMS is to obtain the SoC.
- SoC is a percentage of the remaining usable capacity on a scale from 0 to 100 %.



# Coulomb Counter (CC)

- CC is a 2's compliment counter, which is centered at 0x0000_0000.
- CC updates from battery Open Circuit Voltage (OCV) to reduce integrated error of SoC
- CC counts up when a charge is removed from the battery
  - BMS_CSP is negative relative to BMS_CSN pin
- CC counts down when the battery is being charged
  - BMS_CSP is positive relative to BMS_CSN pin
- BMS_CSP must always be connected to negative side of battery and BMS_CSN must be grounded

# BMS_CSP and BMS_CSN




BMS trace routing rules:

1. BMS_CSP must connect to negative terminal of battery. Don't change polarity!
2. Route traces directly to $R_{sense}$ pad without sharing any power trace!
3. Keep trace resistance (Rp) less than 1 Ω.
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.

# Unit Conversion of CC

- 4-byte CC accumulates the last stored CCADC measurement with an accumulation rate of 1.67847 ms
- LSB of CC is 86.8056 μV

**Question** What is the accumulated value of CC given a fixed current of 1 A going through a sense resistor of 10 mΩ?

**Answer** This will create a measurement value of 10 mV and the CC will increment by 115 decimal bits (= 10 mV/86.8056 μV) each 1.67847 ms.

# CC Reset and Saturation

- CC will reset if:
  - Its value crosses the predefined threshold
  - OCV is updated
- In PM8921 2.0, maximum battery size that could be supported with the Coulomb counter without saturation

| PM8921 2.0 module | | |
|---|---|---|
| Resistor (Ω) | CC LSB (µV) | Max. battery size (Ah) |
| 0.010 | 86.806 | 8.73 |
| 0.015 | 86.806 | 5.82 |
| 0.020 | 86.806 | 4.37 |
| 0.025 | 86.806 | 3.49 |

# CC Threshold

- For the PM8921 2.0 module, the CC size is large enough to avoid saturation, and the threshold should never be exceeded.
- Therefore, the threshold should be set 20% above the battery size as follows:

CC_threshold = (0x7FFF_FFFF)*(1.2*battery_size)/(max_battery_size)

# PM8921 BMS System Outputs

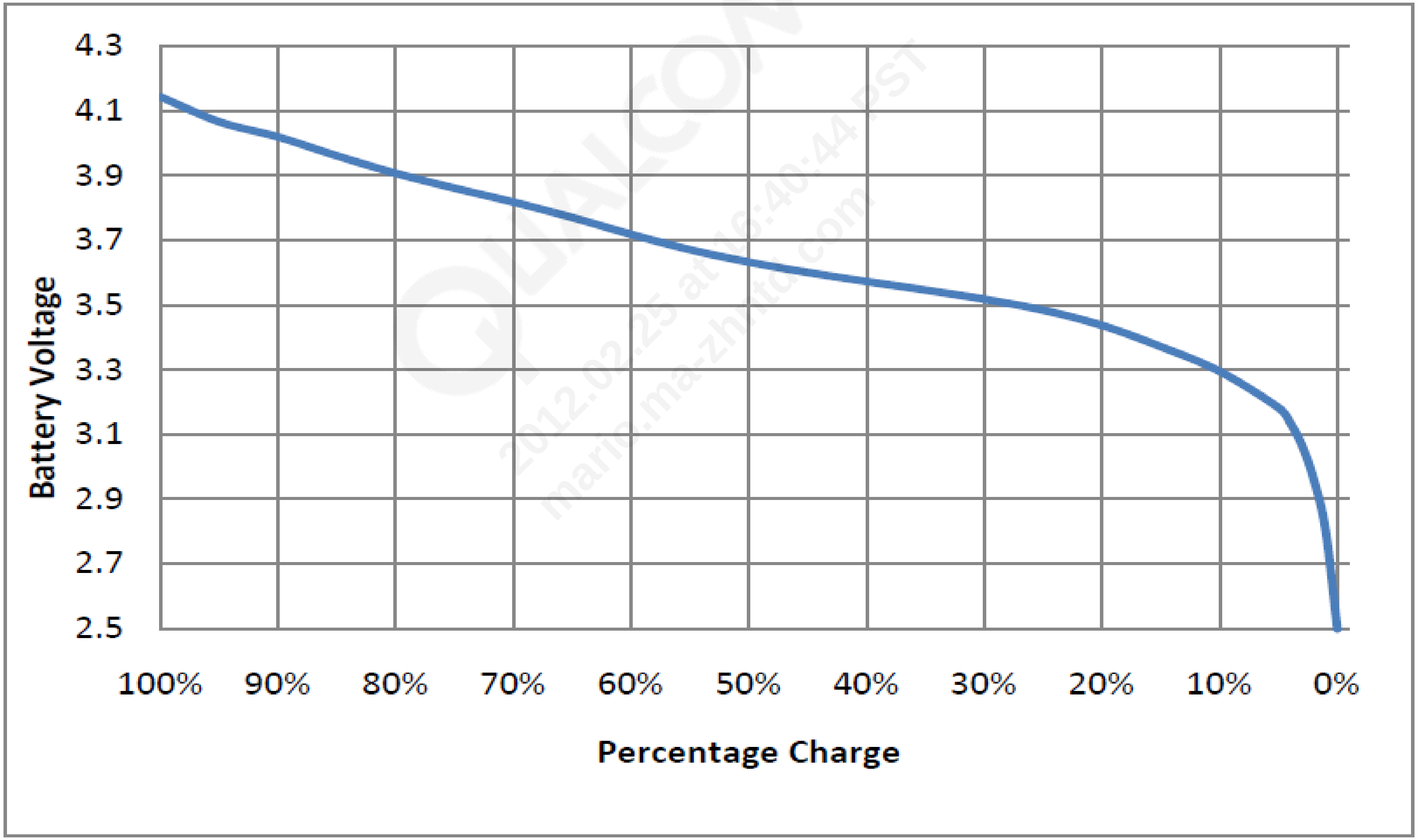
- To determine SoC, the following parameters are needed:
  - Itest − Peak system current
  - Vfailure − Voltage at which point the battery is considered empty by the system
  - FCC − Full charge capacity of the battery
  - CCmAh − Accumulated charger in terms of mAh
  - OCV − Open circuit voltage of the battery
  - Rbat − Battery internal resistance
- To compute Rbat, three parameters are needed:
  - Vbatt for resistance measurement – [VB_1]
  - Vsense for resistance measurement – [VS_1]
  - OCV for resistance measurement – [VB_2]
  - Rbatt = (VB_2-VB_1)*Rsense/VS_1

# Calculation of SoC Metrics

- The Algorithm to compute SoC is:
  - Rbatt = (VB_2-VB_1)*Rsense/VS_1
  - Unusable Charge (UUC) = FCC*Lookup(Rbatt* Itest + Vfail)
  - Battery Percentage Charge (PC) = Lookup(OCV)
  - Remaining Charge (RC) = PC * FCC
  - Remaining Usable Charge (RUC) = RC – CCmAh - UUC
  - SoC = RUC/(FCC-UUC)
  - Value of SoC between 0 and 100. 0 means the battery is fully depleted
- Lookup() − A function that returns the remaining Percentage Charge (PC) given a voltage

# Diagram of Battery Voltage vs Percentage Charge

Percentage Charge = Lookup(OCV)



# PM8921 Nonvolatile BMS Parameters

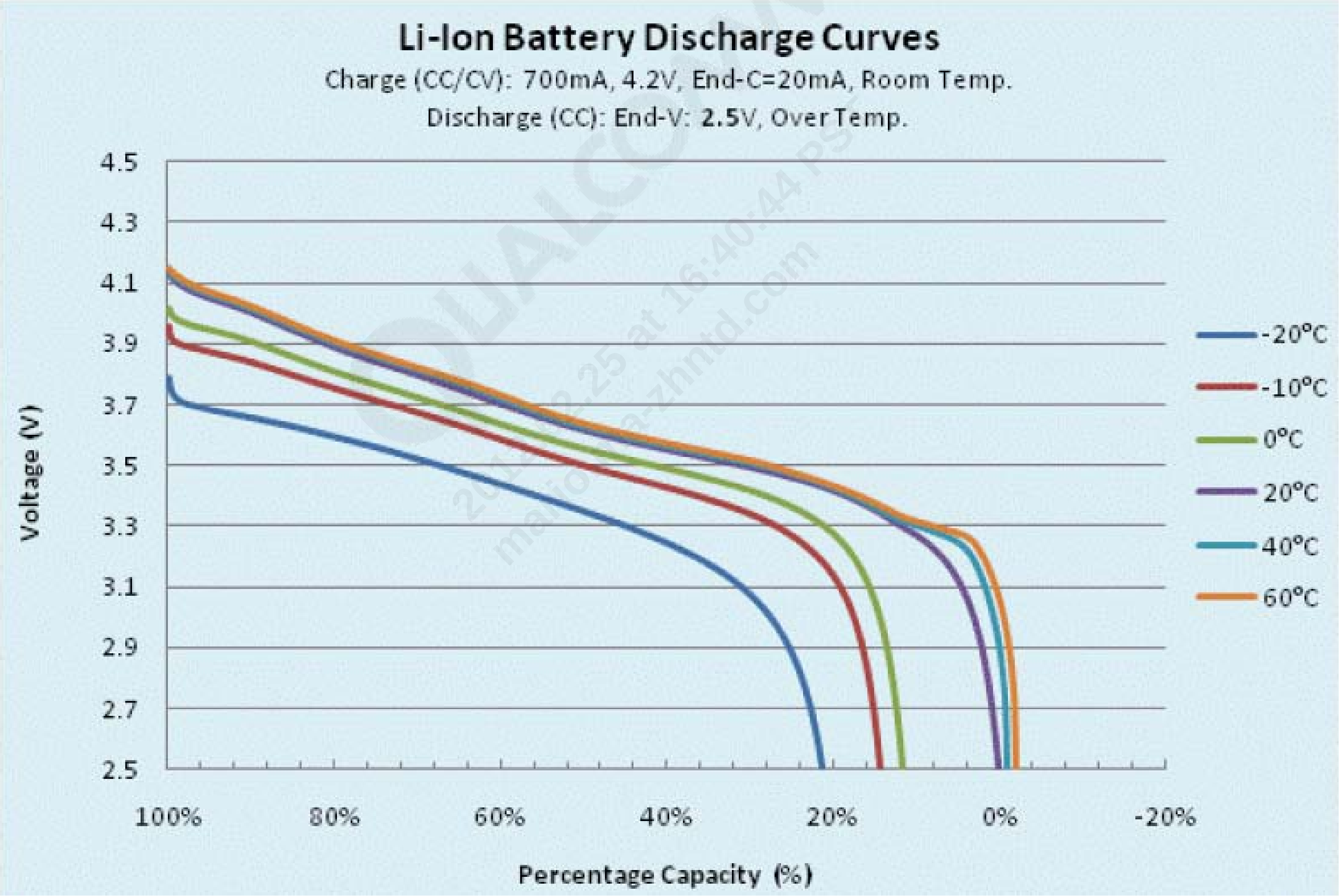
- Some BMS parameters need to be restored after reboot
- A BMS daemon is used to keep the following parameters
  - last_rbatt
  - last_ocv_uv
  - last_chargecycles – Indicator of the battery age
  - last_charge_increase – SoC of the current charge cycle
- Computing of charge cycles

| Charge | Starting SoC | Ending SoC | SoC change |
|---|---|---|---|
| 1 | 22% | 74% | 52% |
| 2 | 35% | 100% | 65% |
| 3 | 55% | 85% | 30% |
| 4 | 17% | 95% | 78% |
| | | Total SoC change = | 225% |
| | | Charge cycles = | 2.25 |

# PM8921 BMS Calibration

- PM8921 provides FCC and PC tables for SoC computing.
- FCCs are affected by two factors:
  - Temperature
  - Charge cycle
- PCs are affected by three factors:
  - Temperature
  - Charge cycle
  - OCV
- Therefore, both FCC (temp,chg_cycle) and PC (temp,chg_cycle,ocv) are needed to be calibrated before SoC is computed.
- The QCT BMS profiling tool will be released at the end of Nov 2011.

# PC vs Temperature Graph



# References

| Ref. | Document | |
|---|---|---|
| ***Qualcomm*** | | |
| Q1 | *Application Note: Software Glossary for Customers* | CL93-V3077-1 |

# Questions?

**https://support.cdmatech.com**