UEFI PMIC Software

User Guide

80-P2484-42 Rev. C

September 7, 2018

Qualcomm
2018-12-26 17:59:17 PST
zk_sw@wingtech.com

Confidential and Proprietary - Qualcomm Technologies, Inc.

NO PUBLIC DISCLOSURE PERMITTED: Please report postings of this document on public servers or websites to DocCtrlAgent@qualcomm.com.

Restricted Distribution: Not to be distributed to anyone who is not an employee of either Qualcomm Technologies, Inc. or its affiliated companies without the express approval of Qualcomm Configuration Management.

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

All Qualcomm products mentioned herein are products of Qualcomm Technologies, Inc. and/or its subsidiaries.

Qualcomm is a trademark of Qualcomm Incorporated, registered in the United States and other countries. Other product and brand names may be trademarks or registered trademarks of their respective owners.

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 Technologies, Inc.
5775 Morehouse Drive
San Diego, CA 92121
U.S.A.

© 2016, 2018 Qualcomm Technologies, Inc. and/or its subsidiaries. All rights reserved.

Revision history

Revision	Date	Description
A	December 2016	Initial release
B	January 2018	Numerous updates were made to the UEFI QTI charger app LA configuration chapter, including the addition of parameters
C	September 2018	Added updates for newer chipsets

Qualcomm
2018-12-26 17:59:17 PST
zk_sw@wingtech.com

Contents

Qualcomm
2018-12-26 17:59:17 PST
zk_w@wingtech.com

Qualcomm
2018-12-26 17:59:17 PST
zk_sw@wingtech.com

Qualcomm
2018-12-26 17:59:17 PST
zk_sw@wingtech.com

1 Introduction

1.1 Purpose

This document provides an overview of unified extensible firmware interface (UEFI) PMIC software drivers, battery management applications, and battery charging configurations.

1.2 Conventions

Function declarations, function names, type declarations, attributes, and code samples appear in a different font, for example, `#include`.

Button and key names appear in bold font, for example, click **Save** or press **Enter**.

1.3 Technical assistance

For assistance or clarification on information in this document, submit a case to Qualcomm Technologies, Inc. (QTI) at https://createpoint.qti.qualcomm.com/.

If you do not have access to the CDMATech Support website, register for access or send email to support.cdmatech@qti.qualcomm.com.

Qualcomm
2018-12-26 17:59:17 PST

2 Overview

The UEFI specification defines a set of platform-independent APIs to enable the following:

- Interaction between the platform firmware and the operating system/OS loader
- Loading, installing, and executing drivers in a pre-OS environment
- Cross-platform portability for UEFI applications

2.1 Protocol

Protocol is a set of software interfaces used for communication between two binary modules. Each protocol must have a specification that includes the protocol GUID and its interface structure.

Protocol driver dependencies

- Defined in the driver INF file
- Loaded and processed during system initialization

Qualcomm
2018-12-26 17:59:17 PST
zk_sw@wingtech.com

Protocol usage

- To use the interfaces from other modules, the module must locate the protocols through GUIDs during module initialization.
- To expose the interface for other modules, the module must install the protocol during module initialization.

Locate a protocol (example)

The following example demonstrates how to locate the protocol:

```
/* Interface for the protocol to access the RTC */
EFI_QCOM_PMIC_PWRON_PROTOCOL  *pmic_pwron;
…
Status = gBS->LocateProtocol(&gQcomPmicPwrOnProtocolGuid, NULL, (VOID**)
&pmic_pwron);
```

Use the located protocol (example)

Run the following commands to use the interface after the protocol is located:

```
EFI_STATUS Status;
…
```

```
Status = pmic_pwron->GetPonPblStatus (0,
EFI_PM_PWRON_PON_PBL_STATUS_XVDD_RB_OCCURRED,&pmicWasBatteryRemoved);
```

2.2 INF file setting

The INF file is used to specify the sources and dependencies for a module.

RealTimeClockLib.inf (example)

Any module requiring a PMIC resource must include the related protocol GUID in the INF file.

```
…
[Protocols]
    gQcomPmicRtcProtocolGuid
    gQcomPmicPwrOnProtocolGuid
…
```

PmicDxe.inf (example)

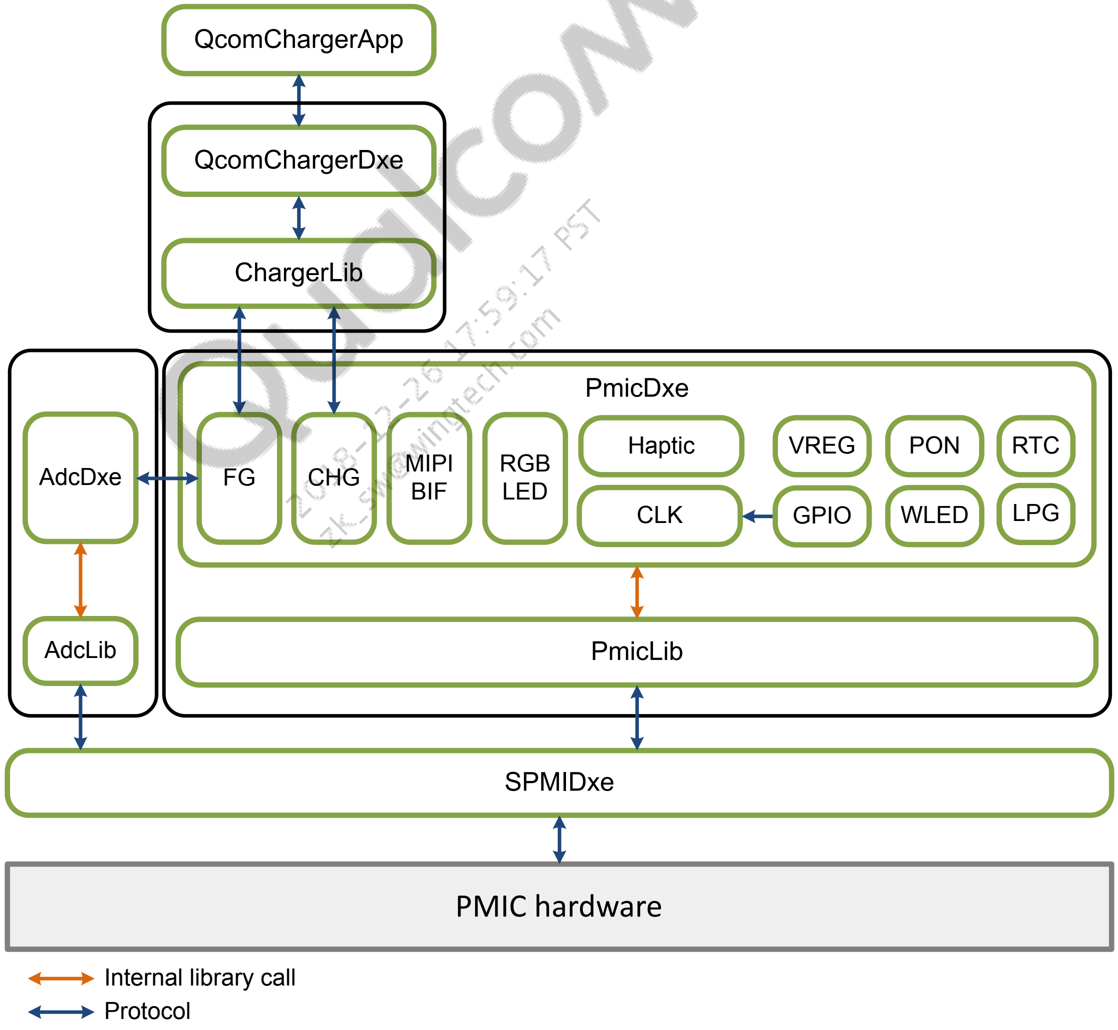
INF enables a module to be linked to the library.

```
…
[LibraryClasses]
  BaseMemoryLib
  PmicLib
…
```

Qualcomm
2018-12-26 17:59:17 PST
zk_sw@wingtech.com

3 UEFI PMIC software drivers

The following figure shows the PMIC software stack:



Qualcomm
2018-12-26 17:59:17 PST
zk_sw@wingtech.com

The following table describes UEFI PMIC drivers:

Driver	Source path	Functionality
AdcDxe	`<src root> \uefi\edk2\QcomPkg \Drivers\AdcDxe`	■ Provides abstraction of ADC functionality through protocols ■ Not platform-specific
AdcLib	`<src root> \uefi\edk2\QcomPkg \Library\AdcLibB`	■ Implements register-level access of ADC functionalities ■ Platform-specific
QcomChargerDxe	`<src root> \uefi\edk2\QcomPkg \Drivers\QcomChargerDxe`	■ Provides abstraction of charging and gauge functionality of the system ■ Interface between Microsoft UEFI charger applications or other high-level UEFI applications, such as Flash ■ Battery management customization
PmicShutdownLib	`<src root> \uefi\edk2\QcomPkg \Library\PmicShutdownLib`	Implementation of boot time and runtime shutdown/reset functionality
PmicDxe	`<src root> \uefi\edk2\QcomPkg \Drivers\PmicDxe`	■ Provides abstraction of most PMIC functionalities through protocols (see PmicDxe driver) ■ Not platform-specific
PmicLib	`<src root> \uefi\edk2\QcomPkg \Library\PmicLib`	■ Implements register-level access of PMIC functionality ■ Platform-specific
SPMIDxe	`<src root> \uefi\edk2\QcomPkg \Drivers\SPMIDxe`	Implementation of SPMI communication with PMIC

Qualcomm
2018-12-26 17:59:17 PST

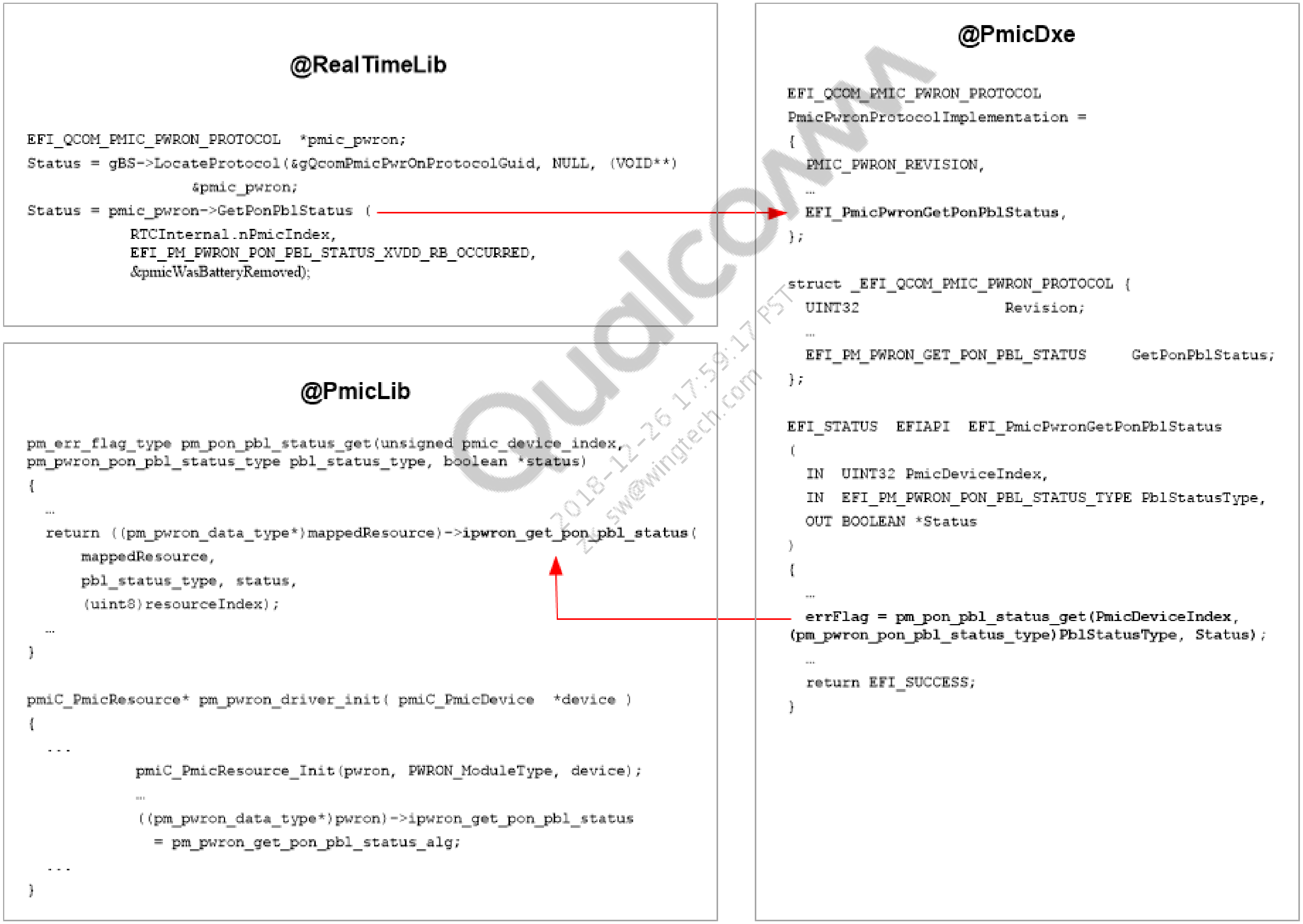
The following table lists the UEFI PMIC protocols. Each header file listed is located in `<src root> \uefi\edk2\QcomPkg\Include\Protocol\`.

PMIC protocol	Header file	Protocol usage
PMIC_CLKBUFF_PROTOCOL	EFIPmicClkBuff.h	PMIC CLK buffer
PMIC_GPIO_PROTOCOL	EFIPmicGpio.h	PMIC GPIO
PMIC_IBB_PROTOCOL	EFIPmicIbb.h	PMIC IBB
PMIC_LAB_PROTOCOL	EFIPmicLab.h	PMIC LAB
PMIC_LPG_PROTOCOL	EFIPmicLpg.h	PMIC LPG
PMIC_MIPIBIF_PROTOCOL	EFIPmicMipiBif.h	PMIC MipiBif
PMIC_MPP_PROTOCOL	EFIPmicMpp.h	PMIC MPP
PMIC_PWM_PROTOCOL	EFIPmicPwm.h	PMIC PWM
PMIC_PWRON_PROTOCOL	EFIPmicPwrOn.h	PMIC PON
PMIC_RGB_LED_PROTOCOL	EFIPmicRgbLed.h	PMIC RGB
PMIC_RTC_PROTOCOL	EFIPmicRTC.h	PMIC RTC
PMIC_SCHG_PROTOCOL	EFIPmicSchg.h	PMIC SCHG
PMIC_FG_PROTOCOL	EFIPmicFg.h	PMIC FG

PMIC protocol	Header file	Protocol usage
PMIC_VIB_PROTOCOL	EFIPmicVib.h	PMIC VIB
PMIC_VREG_PROTOCOL	EFIPmicVreg.h	PMIC VREG
PMIC_PWM_PROTOCOL	EFIPmicPwm.h	PMIC PWM
PMIC_WLED_PROTOCOL	EFIPmicWled.h	PMIC WLED

Qualcomm
2018-12-26 17:59:17 PST
zk_sw@wingtech.com

The following sample code shows the UEFI PMIC driver flow:



@RealTimeLib

```
EFI_QCOM_PMIC_PWRON_PROTOCOL  *pmic_pwron;
Status = gBS->LocateProtocol(&gQcomPmicPwrOnProtocolGuid, NULL, (VOID**)
                 &pmic_pwron;
Status = pmic_pwron->GetPonPblStatus (
           RTCInternal.nPmicIndex,
           EFI_PM_PWRON_PON_PBL_STATUS_XVDD_RB_OCCURRED,
           &pmicWasBatteryRemoved);
```

@PmicLib

```
pm_err_flag_type pm_pon_pbl_status_get(unsigned pmic_device_index,
pm_pwron_pon_pbl_status_type pbl_status_type, boolean *status)
{
  …
  return ((pm_pwron_data_type*)mappedResource)->ipwron_get_pon_pbl_status(
      mappedResource,
      pbl_status_type, status,
      (uint8)resourceIndex);
  …
}

pmiC_PmicResource* pm_pwron_driver_init( pmiC_PmicDevice  *device )
{
  ...
            pmiC_PmicResource_Init(pwron, PWRON_ModuleType, device);
            …
            ((pm_pwron_data_type*)pwron)->ipwron_get_pon_pbl_status
              = pm_pwron_get_pon_pbl_status_alg;
  ...
}
```

@PmicDxe

```
EFI_QCOM_PMIC_PWRON_PROTOCOL
PmicPwronProtocolImplementation =
{
  PMIC_PWRON_REVISION,
  …
  EFI_PmicPwronGetPonPblStatus,
};

struct _EFI_QCOM_PMIC_PWRON_PROTOCOL {
  UINT32                Revision;
  …
  EFI_PM_PWRON_GET_PON_PBL_STATUS     GetPonPblStatus;
};

EFI_STATUS  EFIAPI  EFI_PmicPwronGetPonPblStatus
(
  IN  UINT32 PmicDeviceIndex,
  IN  EFI_PM_PWRON_PON_PBL_STATUS_TYPE PblStatusType,
  OUT BOOLEAN *Status
)
{
  …
  errFlag = pm_pon_pbl_status_get(PmicDeviceIndex,
(pm_pwron_pon_pbl_status_type)PblStatusType, Status);
  …
  return EFI_SUCCESS;
}
```

Qualcomm
2018-12-26 17:59:17 PST
zz.sw@wingtech.com

3.1 Core

3.1.1 No OS-level interrupt support

- UEFI is a single thread OS
- Cannot register the interrupt service routine for PMIC interrupt
- PMIC interrupt can be enabled to detect an event through the latched status bit
- Interrupts enabled in UEFI must be disabled before booting to HLOS to prevent unknown wakeup activities in the HLOS
- Timer or timer callback can be used as alternatives

3.1.2 No XO shutdown or deep sleep

- Not supported due to framework limitation
- Improves power efficiency during UEFI charging
- Low Power mode is supported by reducing the CPU clock speed and number of active cores
- Display is on for a limited amount of time during boot and can be turned on when you press the power and volume keys during UEFI low battery charging

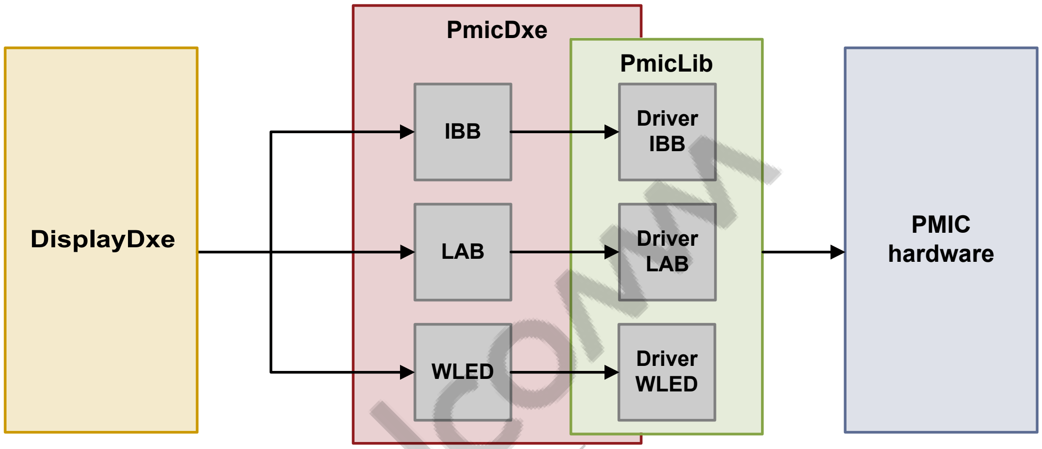
3.1.3 VREG is controlled via NPA/RPM instead of direct register access

- Do not use the following functions to set the VREG register (these functions cause a device crash if misused):
  - PmicVRegProtocol > VregSetLevel
  - PmicVRegProtocol > VregControl
- TZ PMIC SPMI permissions become active when the time device boots up to UEFI.

Qualcomm 2018-12-06 17:57:17 PST ck_swewingtech.com

3.2 User interface

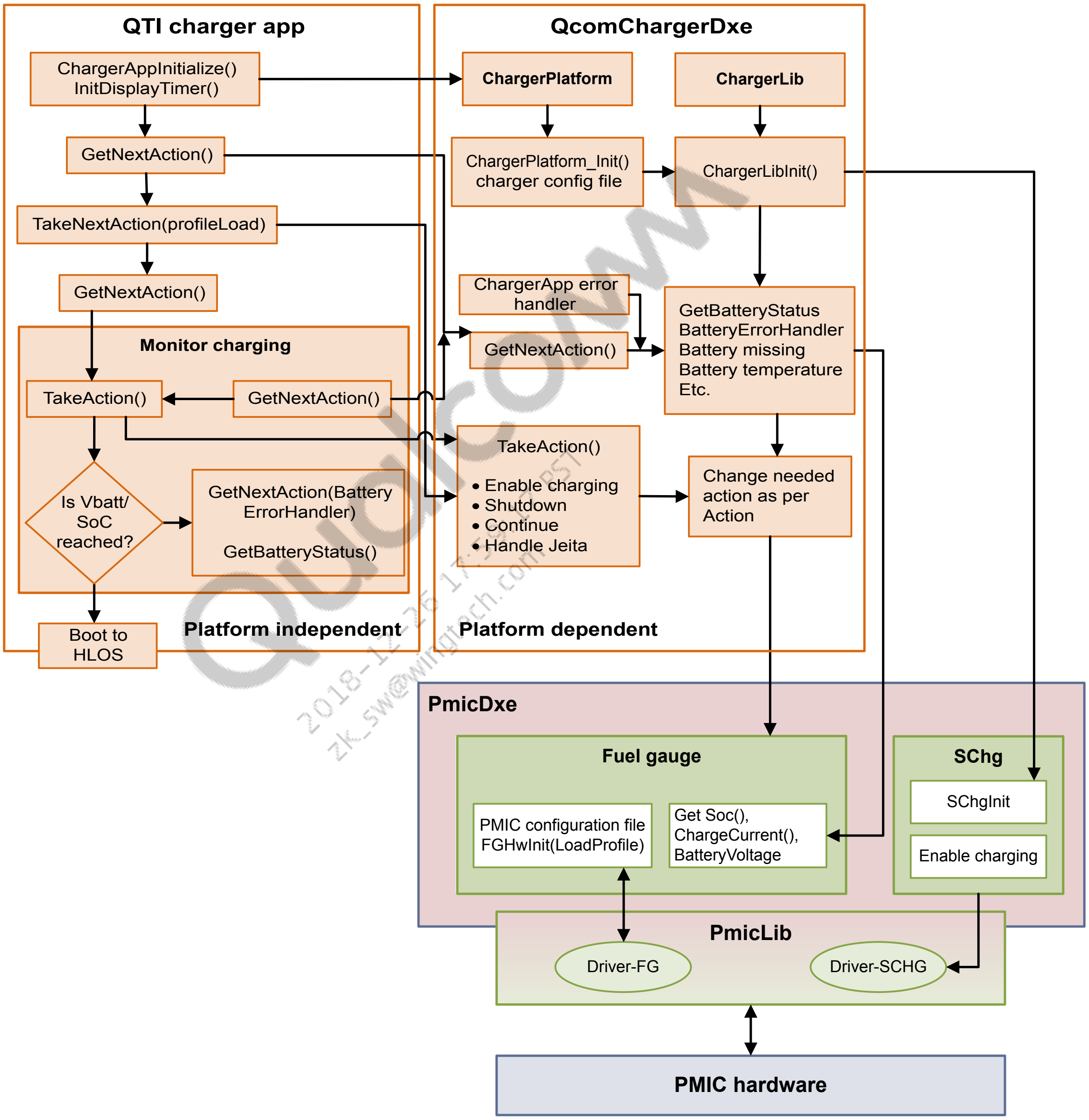
The following figure shows the PMIC drivers used to configure IBB, LAB, and WLED for the display:



Qualcomm
2018-12-26 17:59:17 PST
zk_sw@wingtech.com

3.3 UEFI charger app

The following is a diagram of the UEFI charger app design:



2018-12-26 17:59:51 PST
zk_sw@wingtech.com

3.3.1 PmicDxe driver

The PmicDxe driver implements functions to program the charger and fuel gauge (FG) hardware. It also implements such functions as parsing/load battery profile, FG boot sequence, AICL rerun, and so on.

3.3.2 ChargerLib

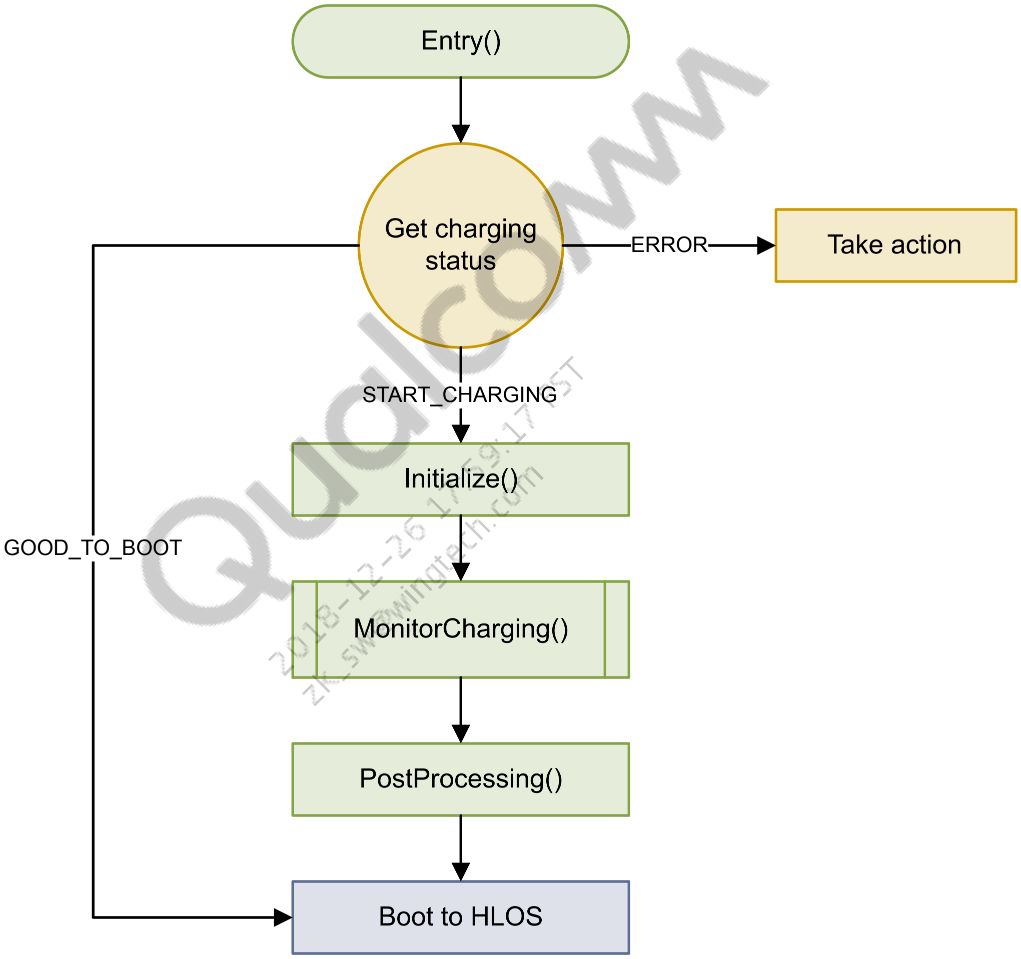
The platform-dependent ChargerLib provides the following functions:

- Enables/disables charging
- Sets charger maximum battery current
- Float voltage (FV)
- Checks charger source
- Gets battery state of charge (SoC), voltage, and current
- Handles battery errors and hardware Jeita AFP thresholds

Qualcomm
2018-12-26 17:59:17 PST
zk_sw@wingtech.com

3.3.3 QTI charger app

The QTI charger app supports voltage-based and SoC-based threshold charging. The QTI charger app calls ChargerLib to determine if it is OK to boot to HLOS or whether to stay in UEFI charging until the threshold is reached. The QTI charger app has one main loop that keeps the application running, checking if it is OK to exit. The following figure shows the QTI charger app flow:



Qualcomm 2018-12-26 17:59:17 PST zk_sw@wingtech.com

The charger has a timer that performs the following tasks:

- Turns the display ON/OFF
- Displays the charging/battery icon
- Runs the following in a loop every 3 sec until threshold is reached:
  - Check battery status
  - Manage charging
  - Error handling

Additional support includes safety features, such as thermal mitigation, WiPower, etc.

The following table describes the QTI charger app functions:

Source file	Function	Description
QcomChargerApp.c	`Entry()`	Initializes the QTI charger app, starts the charging loop, and cleans up after exiting the charging loop
	`Initialize()`	Sets the charging parameters and starts the display timer
	`MonitorCharging()`	Charging loop that monitors battery charging status until battery status is at a good enough charge to boot to HLOS
	`PostProcessing()`	Called after exiting the charging loop; closes all events and timers, and decides whether to boot to HLOS, stay in UEFI, or shut down
	`DeInitialize()`	Performs required exit actions before leaving the app
QcomChargerApp-EventHandler.c	`KeyPressEventHandler()`	Turns on the display, starts the display timer, and exits LPM mode when the volume up/ down, power key, or home key is pressed
	`KeyPressControl()`	Registers or unregisters volume up, volume down, and home key press event callback with the keypad driver
	`HandleLPMClock()`	Signals the clock driver to enter or exit LPM
	`HandleLPMDisplay()`	Signals the display driver to enter or exit LPM to turn the display ON or OFF
	`EnterLPM()`	Called when the display is turned off; sends an enter LPM signal to clock and display drivers
	`ExitLPM()`	Called when the display is turned on by key press; sends an exit LPM signal to the clock and display drivers
	`AnimImgTimer()`	Animates image display during charging (display ON)
	`DisplayTimerEvent()`	Displays the battery image during charging (display ON)

Qualcomm 2018-12-26 17:59:17 PST zk_sw@wingtech.com

4 UEFI QTI charger app LA configuration

The UEFI QTI charger app handles the following:

- Charges the battery to a specified level
- Determines if the battery is OK to boot to HLOS without UEFI charging

See QTI charger app for an overview of the QTI charger app.

4.1 QTI charger app configuration file

The QTI charger app includes a build-time configuration file with parameters for charger/FG management.

See QTI charger app configuration file for LA (for PM8998) for an example file.

The configuration file is located in the following:

`/QcomPkg/Drivers/QcomChargerDxe/QcomChargerConfig_VbattTh_<chipset>.cfg`

`<chipset>` corresponds to the applicable chipset product. For example: `/QcomPkg/Drivers/QcomChargerDxe/QcomChargerConfig_VbattTh_SDM845.cfg`.

The configuration file must be renamed QcomChargerCfg.cfg to be used.

NOTE For newer chipsets (SM8150/SM8160), the file is located in `\boot_images\QcomPkg\SDMPkg\<chipset>\Settings\Charger\QcomChargerConfig_VbattTh_<chipset>.cfg`

Qualcomm

4.2 Debug overwrite feature

The debug overwrite feature enables debug logs and overwrites UEFI configuration at runtime. Overwriting configuration at runtime makes it unnecessary to rebuild the core. On debug builds, the UEFI QTI charger app is built during initialization.

This feature only works on debug builds.

4.2.1 Enable the debug overwrite feature with Flash tools FV

The QComChargerCfg.cfg file can be copied via mass storage application from the UEFI boot device selection (BDS) menu to mount the LogFS drive on which to copy the file. The FV tool must be flashed to enable the UEFI BDS menu.

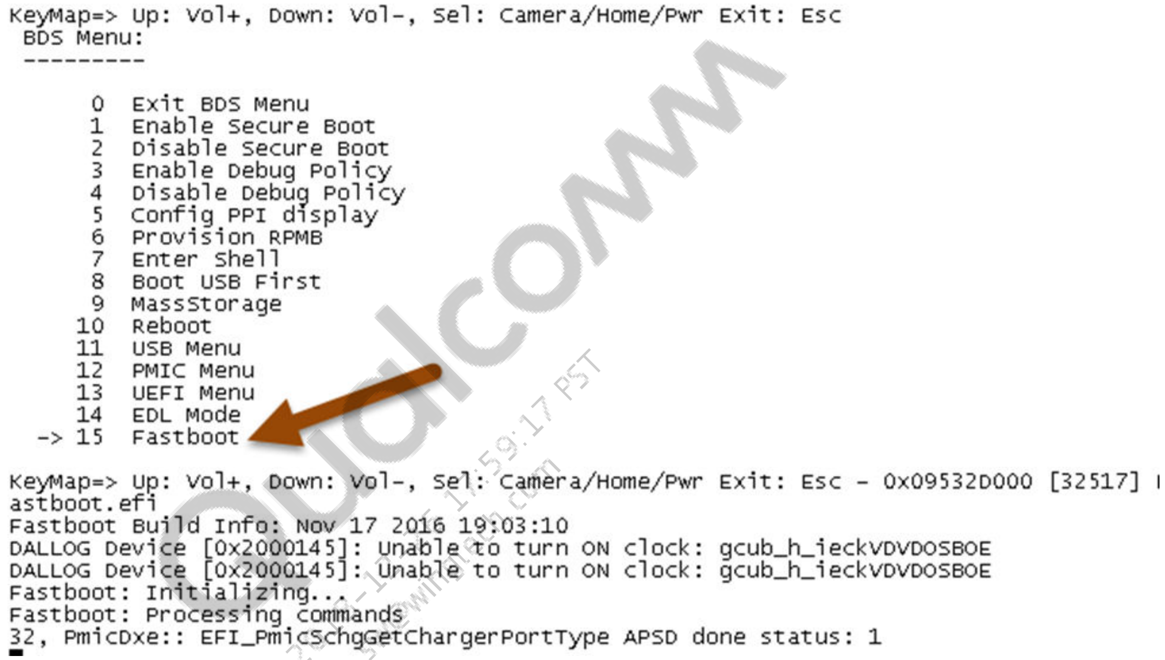
The tools.fv file is located in the following metabuild folder:

```
..\boot_images\QcomPkg\QcomToolsPkg\Bin\QcomTools\DEBUG
```

1. Run the following command:

   ```
   fastboot flash toolsfv ..\boot_images\QcomPkg\QcomToolsPkg\Bin\QcomTools
   \DEBUG\tools.fv
   ```

2. During boot, hold down the **volume** key to display the BDS menu.
3. Use the **volume down** key to scroll down the menu, and press the **Home/Power** key to select Fastboot.



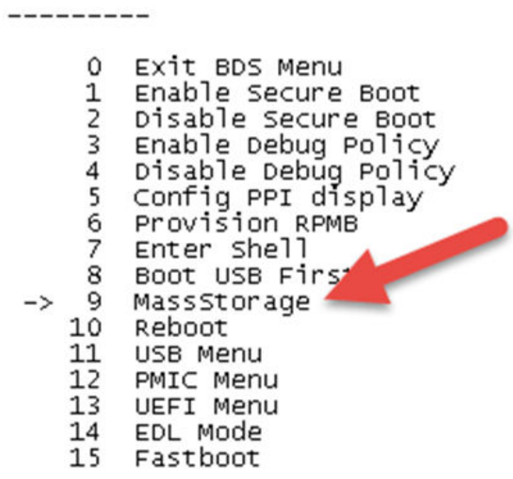
```
KeyMap=> Up: Vol+, Down: Vol-, Sel: Camera/Home/Pwr Exit: Esc
 BDS Menu:
 ---------

        0   Exit BDS Menu
        1   Enable Secure Boot
        2   Disable Secure Boot
        3   Enable Debug Policy
        4   Disable Debug Policy
        5   Config PPI display
        6   Provision RPMB
        7   Enter Shell
        8   Boot USB First
        9   MassStorage
       10   Reboot
       11   USB Menu
       12   PMIC Menu
       13   UEFI Menu
       14   EDL Mode
    -> 15   Fastboot

KeyMap=> Up: Vol+, Down: Vol-, Sel: Camera/Home/Pwr Exit: Esc - 0x09532D000 [32517] 
astboot.efi
Fastboot Build Info: Nov 17 2016 19:03:10
DALLOG Device [0x2000145]: Unable to turn ON clock: gcub_h_ieckVDVDOSBOE
DALLOG Device [0x2000145]: Unable to turn ON clock: gcub_h_ieckVDVDOSBOE
Fastboot: Initializing...
Fastboot: Processing commands
32, PmicDxe:: EFI_PmicSchgGetChargerPortType APSD done status: 1
```

Qualcomm

4. Reboot the device, press and hold down the **volume** key to open the UEFI BDS menu, and press the **volume down** key to select MassStorage.

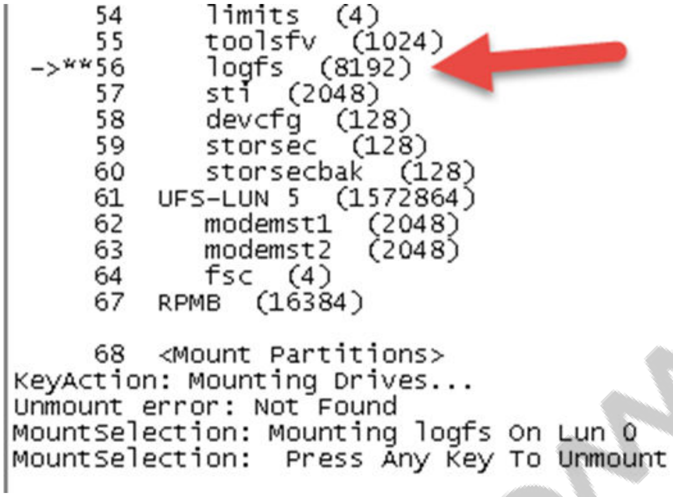


```
 ---------

        0   Exit BDS Menu
        1   Enable Secure Boot
        2   Disable Secure Boot
        3   Enable Debug Policy
        4   Disable Debug Policy
        5   Config PPI display
        6   Provision RPMB
        7   Enter Shell
        8   Boot USB First
    ->  9   MassStorage
       10   Reboot
       11   USB Menu
       12   PMIC Menu
       13   UEFI Menu
       14   EDL Mode
       15   Fastboot
```

5. Press the **Home/Power** key to launch the UEFI MassStorage app.

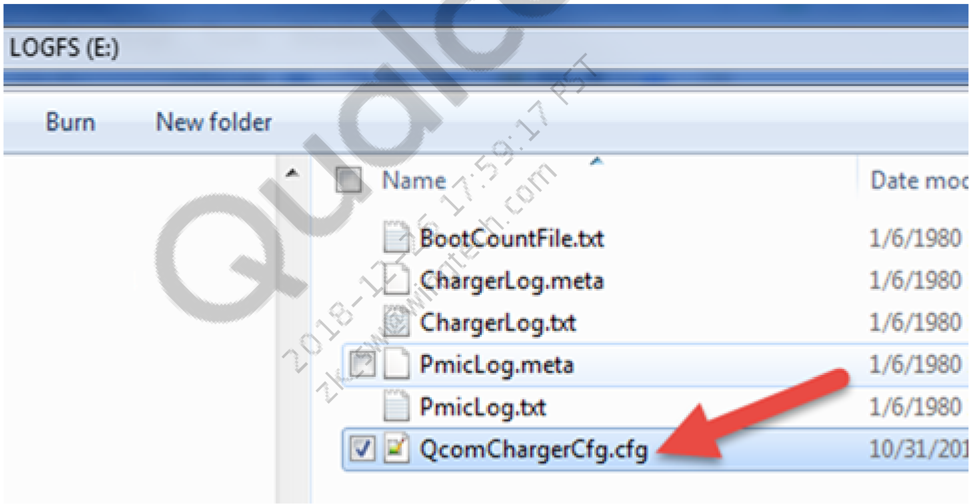
6. Mount the LogFS partition.



```
      54     limits  (4)
      55     toolsfv  (1024)
  ->**56     logfs  (8192)
      57     sti  (2048)
      58     devcfg  (128)
      59     storsec  (128)
      60     storsecbak  (128)
      61  UFS-LUN 5  (1572864)
      62     modemst1  (2048)
      63     modemst2  (2048)
      64     fsc  (4)
      67  RPMB  (16384)

      68  <Mount Partitions>
KeyAction: Mounting Drives...
Unmount error: Not Found
MountSelection: Mounting logfs On Lun 0
MountSelection:  Press Any Key To Unmount
```

7. Copy the QcomChargerCfg.cfg file to the LogFS partition root level with the modified configuration.



Name	Date moc
BootCountFile.txt	1/6/1980
ChargerLog.meta	1/6/1980
ChargerLog.txt	1/6/1980
□ PmicLog.meta	1/6/1980
PmicLog.txt	1/6/1980
☑ QcomChargerCfg.cfg	10/31/201

Qualcomm
2018-12-25 17:59:17 PST

8. Remove LogFS from the task bar.

9. Reboot the device.

The new configuration takes effect and the modified configuration is listed in the LogFS configuration file, that is, the configuration file changes to enable the charger.

4.3 Configuration quick reference table

The following table describes configuration parameters:

Parameter	Description	Default	Dependency	Data format
Threshold charging				
SocBasedBoot	Sets the thresholds in QTI charger app to allow boot when the configured threshold is met	FALSE	LoadBatteryProfile	Boolean
LoadBatteryProfile	Loads the battery profile for battery SoC accuracy	TRUE	DispSignOfLifeMaxThresholdMv	Boolean
DispSignOfLifeMaxThresholdMv	Displays the sign of life during battery profile load, providing a user indication for cold boot	3700 mV	LoadBatteryProfile	mV
FgCondRestart	Decides if the FG must be restarted to allow battery accuracy	TRUE	LoadBatteryProfile	Boolean
ChargerLedConfig	Configures the charger LED status	1	None	Decimal
EnShipMode	If enabled, ChargerApp gets variable value and, if found set, puts device into ShipMode at UEFI level	FALSE	'ShipModeVarStr = ShipMode' Needs UEFI variable service to be working upon reboot	Boolean
Debug				
PrintChargerAppDbgMsg	If TRUE, enables QTI charger app logs and respective pmic.dxe charger logs, and displays them on UART	FALSE	None	Boolean
PrintChargerAppDbgMsgToFile	If TRUE, enables QTI charger app file logs and respective pmic.dxe charger logs, and saves them to a file	FALSE	LogFS partition availability	Boolean
FileLoggingDbgLevelMask	Debug level mask (in hex) for file logging	80000042	None	Bit flag
EnableChargerFGDump	If TRUE, enables PMIC charger and FG peripheral dumps	FALSE	PrintChargerAppDbgMsg PrintChargerAppDbgMsgToFile	Boolean

Qualcomm
2018-12-26 17:59:17 PST
zk_wa@wingtech.com

Parameter	Description	Default	Dependency	Data format
FG and FG debug				
BatteryIdTolerance	Reads the battery ID to load the battery profile and set a tolerance limit	8%	LoadBatteryProfile	8% ± on current battery ID reading
DumpSram	Enables QTI charger app FG SRAM dumps	FALSE	PrintChargerAppDbgMsg PrintChargerAppDbgMsgToFile	Boolean
DumpSramStartAddr	SRAM dump start address (values in decimal)	0	DumpSram	Decimal
DumpSramEndAddr	SRAM dump end address (values in decimal)	124	DumpSram	Decimal
DumpSramDuration	Dump SRAM contents timer duration in seconds	90 sec	DumpSram	Seconds
Required initially				
BootToHLOSThresholdInMv	Configures the QTI charger app threshold to allow boot	3600 mV	None	mV
OsStandardBootSocThreshold	Configures the QTI charger app minimum threshold to allow boot	7	LoadBatteryProfile	1 to 100% SoC
BattVoltLimHighDelta	Enables delta FV and current limit to charge the battery	30 mV	None	mV
ChgFvMax	Enables battery FV	4350 mV	None	mV
ChgFccMax	Enables fast charging current	2000 mA	None	mA
ChargingTermCurrent	Enables charger termination current to declare 100% SoC	200 mA	None	mA
ConservChgFvDelta	Enables maximum charger delta FV maximum for unknown battery configurations	200 mV	None	mV
BATT_THERM coefficients	Enables battery thermal coefficients to read the battery temperature accuracy	BATT_THERM coefficients	ProgramBattThermCoeffs=TRUE	Hexidecimal

Qualcomm
2018-12-26 17:59:17 PST
zk_sw@wingtech.com

Parameter	Description	Default	Dependency	Data format
BATT_THERM configs	Enables battery thermal bias wait and ground select configs; supported configs parameters are: ■ BattThermBiasWait ■ BattThermGndSel	BATT_THERM configs	None	BATT_THERM configs
AUX_THERM coefficients	Enables auxiliary thermal coefficients to adjust voltage to the temperature mapping	AUX_THERM coefficients	ProgramAuxThermCoeffs=TRUE	Hexidecimal
AUX_THERM configs	Enables battery auxiliary thermal bias wait and ground select configs; supported configs parameters are: ■ AuxThermBiasWait ■ AuxThermGndSel	AUX_THERM configs	None	AUX_THERM configs
Device skin and charger hot thresholds	Enables configuring device skin and charger hot thresholds	Device skin and charger hot thresholds	ProgramSkinAndChargerHot Threshold = TRUE	Hexidecimal
Charger_THERM source configs	Enables/disables charger thermal source	■ SkinTempSrc = TRUE ■ DieTempSrc = TRUE ■ DieTempCompSrc = TRUE	None	BOOLEAN
EmergencyShutdownVbatt	Configures the device emergency shutdown limit	3200 mV	None	mV
EnableChargerWdog	Enables the charger watchdog to safeguard unintentional charging if the software gets stuck	TRUE	None	Boolean
VBtEmpty threshold	Configures the low battery voltage threshold for the SoC empty interrupt	2800 mV	None	mV
VBattEstDiffThreshold	Configures the estimated voltage difference threshold to restart the FG if the threshold difference is higher	30 mV	FgCondRestart	mV
RConnComp compensation resistance	Configures RConn compensation resistance; value in mOhms - range is ±100 mOhms	0 ohms	None	Ohms

Qualcomm

Parameter	Description	Default	Dependency	Data format
Battery error handling				
DebugBoardBatteryIdMin and DebugBoardBatteryIdMax	Specifies the debug board battery ID range	2000 to 14000	DebugBoardBehavior	Ohms
SmartBatteryIdMin and SmartBatteryIdMax	Specifies the smart battery ID range	240000 to 450000	None	Ohms
RegularBatteryIdMin and RegularBatteryIdMax	Specifies the regular battery ID range	15000 to 137000	None	Ohms
UnknownBatteryBehavior	Defines unknown battery behavior detects if the battery ID is within the specified range	BattID < 2000 and ID > 450000	Battery error handling configuration	Ohms
DebugBoardBehavior	Defines debug board battery behavior	2	Battery error handling configuration	Decimal
BattMissingCfg	Configures the battery missing detection behavior	0	None	Decimal
Jeita				
Jeita zones	Enables specified Jeita zones	Jeita configuration	None	Celsius
JeitaCcCompCfg	Enables configuring Jeita charge current compensation when device is within the battery temperature soft limit for hardware Jeita	1000 mA	None	mA
JeitaFvCompCfg	Enables configuring Jeita charge voltage compensation when device is in battery temperature soft limit for hardware Jeita	105 mV	None	mV
NoChargeAndWait	Configures device behavior for temperatures outside of the charging range but within the operating range	TRUE	None	Boolean
WiPower				
WiPowerSupported	Enables configuring WiPower support for the QTI charger app charging device	TRUE	None	Boolean
DCInBootToHLOSThresholdInMv	Enables configuring the WiPower threshold for QTI charger app charging	3600 mV	None	mV

Qualcomm 2017-12-26 17:59:17 PST zk_sw@wingtech.com

Parameter	Description	Default	Dependency	Data format
SuspendDCIn	Enables configuring suspended DCIN behavior when WiPower is enabled	FALSE	None	Boolean
Thermal				
SWThermalMitigationEnable	Configures thermal safety mitigation in the QTI charger app charging device	FALSE	None	Boolean
TsensTimeoutMins	Configures a thermal safety timer when the device is in the thermal zone and not in charger wait state	30 min	None	Minutes
Tsens limits or zone	Configures thermal safety zones or limits	Tsens limits or zone	None	Celsius
See QTI charger app configuration file for LA (for PM8998) for a QTI charger app LA example file.				

Qualcomm
2018-12-26 17:59:17 PST
zk_sw@wingtech.com

4.4 Threshold charging configuration

The following sections describe threshold charging configuration parameters. See QTI charger app configuration file for LA (for PM8998) for a QTI charger app LA example file.

4.4.1 SocBasedBoot

This parameter sets the thresholds in the QTI charger app to allow boot when the configured threshold is met. The threshold can be voltage- or SoC-based.

- Default value – FALSE (indicates the voltage-based thresholds charging)
- To allow accurate SoC estimates during charging, this parameter must be to set to the LoadBatteryProfile configuration.

4.4.2 LoadBatteryProfile

This parameter loads the battery profile for battery SoC accuracy. If enabled, the QTI charger app loads profile data to the FG first.

- Default value – TRUE (indicates voltage-based thresholds charging)
- This configuration might add a delay of ~1.5 sec to boot while the profile loads and the SoC estimate is calculated. DispSignOfLifeMaxThresholdMv configuration shows sign of life while loading the battery profile. This delay/wait only applies to cold boot or battery removal.
- The FG SRAM profile integrity status dedicated register contains the profile load and FG restart status.

Qualcomm 2018-12-26 17:55 PST zk_sw@wingtech.com

4.4.3 DispSignOfLifeMaxThresholdMv

This parameter displays the sign of life during battery profile load, providing a user indication for cold boot. If the LoadBatteryProfile parameter is enabled, the QTI charger app shows sign of life and loads the profile first.

- Default value – 3700 mV (used to decide on displaying image)
- Dependency – LoadBatteryProfile must be set for an accurate SoC estimate during charging.

4.4.4 FgCondRestart

This parameter decides if the fuel gauge must be restarted to allow battery accuracy for the following condition:

```
If abs(Vbatt_Estimate_diff ) > Vbatt_Estimate_diff_threshold
```

- Default value – TRUE
- Dependency – LoadBatteryProfile must be set.

4.4.5 ChargerLedConfig

This parameter configures the charger LED status. If enabled, the LED turns off after threshold charging is complete, that is, when the device boots to HLOS. Supported values are:

- 0 – Disable
- 1 (default) – Solid during charging
- 2 – LED blinks during charging

4.4.6 EnShipMode

If this parameter is enabled based on ‘ShipModeVarStr = ShipMode’, ChargerApp gets the variable value and, if found set, puts the device into ShipMode at the UEFI level.

- Default value – FALSE
- Dependency – ShipModeVarStr must be set to ShipMode by HLOS or other tool/module used by OEM to set the device to ShipMode in factory. Once set, the device is put into ShipMode when rebooted (until end user restarts in mission mode). This parameter needs UEFI variable service to be working upon reboot.
- Use ShipMode string to query variable status.

Qualcomm

4.5 Debug configuration

The following sections describe debug configuration parameters, which only work on nonproduction builds. See QTI charger app configuration file for LA (for PM8998) for a QTI charger app LA example file.

4.5.1 PrintChargerAppDbgMsg

If TRUE, this parameter enables QTI charger app logs and respective pmic.dxe charger logs, and displays them on UART.

- Default value – FALSE (excludes extensive charging logs and only displays battery status information on UART for general users)
- Example charger log

```
 -----------------------------
 - 0x09C0AD000 [16004] QcomChargerApp.efi
QcomChargerApp:: QcomChargerApp_MonitorCharging
TimeStamp,StateOfCharge,Voltage,ChargeCurrent,Temp
16, ChargerApp:: Battery Status 1,3536,206,25
Waiting for 3 sec
0x2B, ChargerApp: Battery Status 17,3732,-1072,24
Waiting for 3 sec
```

4.5.2 PrintChargerAppDbgMsgToFile

If TRUE, this parameter enables QTI charger app file logs and respective pmic.dxe charger logs, and saves them to a file.

- Default value – FALSE (indicates excludes extensive charging logs and only displays battery status information on UART for general users)
- If file log configuration is enabled, the chargerlog.txt and pmiclog.txt files are generated in the debug LogFS 8 MB partition. When the file size reaches its limit, this configuration acts as a circular buffer and starts overwriting old logs.
- The chargerlog.txt file has the charger and boot configuration. The pmiclog.txt file has FG and charger-related information, for example, SRAM dumps.

4.5.3 FileLoggingDbgLevelMask

This parameter provides a debug level mask (in hex) for file logging.

- Default value – 80000042
- Refer to DebugLib.h

4.5.4 EnableChargerFGDump

If TRUE, This parameter enables PMIC charger and FG peripheral dumps. Dumps also occur during charger app initialization and exit. Dumps are initiated if any key is pressed while charging.

- Default value – FALSE
- Dependency – PrintChargerAppDbgMsg charger logs must be enabled.

4.6 FG and FG debug configuration

The following sections describe FG and FG debug configuration parameters. See QTI charger app configuration file for LA (for PM8998) for a QTI charger app LA example file.

4.6.1 BatteryIdTolerance

This parameter reads the battery ID to load the battery profile and set a tolerance limit. If the battery ID falls into tolerance range, only the respective profile with that battery ID is flashed to the FG. Otherwise, the default profile is loaded from the battery profile file, which is the first profile.

- Default value – 8%
- QTI charger app includes eight profile supports

Qualcomm
2018-12-20 17:59:17 PST
zk_[illegible]@wingtech.com

4.6.2 DumpSram

This parameter enables QTI charger app FG SRAM dumps. If enabled (TRUE), the QTI charger app periodically dumps SRAM to get debug information from hardware FG algorithms.

- Default value – FALSE
- Dependency – PrintChargerAppDbgMsg charger logs must be enabled.

4.6.3 DumpSramStartAddr

This parameter is the SRAM dump start address (values in decimal). Note the starting address differences in PMIC families in the cfg examples.

4.6.4 DumpSramEndAddr

This parameter is the SRAM dump end address (values in decimal). Note the starting address differences in PMIC families in the cfg examples.

4.6.5 DumpSramDuration

This parameter is the dump SRAM contents timer duration in seconds. The default value is 90 sec.

4.7 Initial required charger configuration

The following sections describe initial required charger configuration parameters. See QTI charger app configuration file for LA (for PM8998) for a QTI charger app LA example file.

4.7.1 BootToHLOSThresholdInMv

This parameter configures the QTI charger app threshold to allow boot.

- Default value – 3600 mV
- This threshold is also used for unsupported batteries or battery emulators.

4.7.2 OsStandardBootSocThreshold

This parameter configures the QTI charger app minimum threshold to allow boot.

- Default value – 7 % SoC
- Dependencies
    - SocBasedBoot = TRUE
    - LoadBatteryProfile = TRUE
- This threshold is not used in unsupported batteries or battery emulators.

Qualcomm
2018-12-2… 17:53:17 PST
zk_swimingtec….com

4.7.3 BattVoltLimHighDelta

This parameter enables delta FV to charge the battery. The default value is 30 mV.

4.7.4 ChgFvMax

This parameter enables battery FV. The default value is 4350 mV.

4.7.5 ChgFccMax

This parameter enables fast charging current. The default value is 2000 mA.

4.7.6 ChargingTermCurrent

This parameter enables charger termination current to declare 100% SoC. The default value is 200 mA.

4.7.7 ConservChgFvDelta

This parameter enables maximum charger delta FV maximum for unknown battery configurations. The default value is 200 mV.

4.7.8 BATT_THERM coefficients

This parameter enables battery thermal coefficients to read the battery temperature accurately. This is picked up as per ThermBias value per device/battery and initial values given are formatted with half-float encoding. If these values must be updated, open a Salesforce case with Qualcomm to ensure the correct coefficients are used. Refer to the BATT_THERM master beta coefficient table in *Understanding PMI8998 Fuel Gauge* (80-VT310-138).

- Default values
    - BattThermC1 = A1
    - BattThermC2 = 50
    - BattThermC3 = FF

  NOTE Coefficient configuration is included in the battery profile for newer PM8150 families. The previously mentioned configuration setup is no longer supported.

- Values are based on the following ThermBias and pull-up resistor value:

  ```
  BattThermHalfRangeInC = 25
  ```

- Dependency – ProgramBattThermCoeffs = TRUE

Qualcomm
12-26 17:59:17 PST

4.7.9 BATT_THERM configs

This parameter enables battery thermal bias wait and ground select configs.

- Supported values
  - BattThermBiasWait
    - 0 = 0 ms
    - 1 = 1 ms
    - 2 = 4 ms
    - 3 = 12 ms
    - 4 = 20 ms
    - 5 = 40 ms
    - 6 = 60 ms
    - 7 = 80 ms
  - BattThermGndSel
    - TRUE = Thermistor is located on the battery pack
    - FALSE = Thermistor on the PCB (skin temp)
- Default values
  - BattThermBiasWait = 4
  - BattThermGndSel = TRUE
- Dependency – None

Qualcomm
2018-12-26 17:59:17 PST
zk_sw@wingtech.com

4.7.10 AUX_THERM coefficients

This parameter enables auxiliary thermal coefficients to adjust voltage to the temperature mapping. Temperature mapping is based on the beta 3435 of the thermistor in use and the associated temperature to 50% ratio (dependent on the pull-up value). Refer to the AUX_THERM master beta coefficient table in *Understanding PMI8998 Fuel Gauge* (80-VT310-138).

- Default values
  - AuxThermC1 = BF
  - AuxThermC2 = 36
  - AuxThermC3 = FF

  NOTE Coefficient configuration is in the XBL core for the PM8150 family.

- Values are based on the following ThermBias and pull-up resistor value:

```
AuxThermHalfRangeInC = 25
```

- Dependency – ProgramAuxThermCoeffs = TRUE

4.7.11 AUX_THERM configs

This parameter enables battery auxiliary thermal bias wait and ground select configs.

- Supported values
  - AuxThermBiasWait
    - 0 = 0 ms
    - 1 = 1 ms
    - 2 = 4 ms
    - 3 = 12 ms
    - 4 = 20 ms
    - 5 = 40 ms
    - 6 = 60 ms
    - 7 = 80 ms
  - AuxThermGndSell
    - TRUE = Thermistor is located on the battery pack
    - FALSE = Thermistor on the PCB (skin temp)
- Default value
  - AuxThermBiasWait = 4
  - AuxThermGndSel = FALSE
- Dependency – None

NOTE Coefficient configuration is in the XBL core for the PM8150 family.

Qualcomm
2018-12-26 17:59:17 PST
zk_sw@wingtech.com

4.7.12 Device skin and charger hot thresholds

This parameter enables device skin and charger hot threshold configuration. Device skin temperature is usually on device display. Charger hot thresholds are used for thermal mitigation via intelligent negotiation for optimum voltage (INOV).

- Default values
  - DeviceSkinHotInC = 80
  - DeviceSkinTooHotInC = 90
  - ChargerHotInC = 80
  - ChargerTooHotInC = 90

- Default values (PM8950)
  - ChargerHotInc = 80
  - ChargerTooHotInc = 90
- Dependency – ProgramSkinAndChargerHotThreshold must be TRUE.

4.7.13 Charger_THERM source configs

This parameter enables/disables the charger thermal source.

- Default values
  - SkinTempSrc = TRUE
  - DieTempSrc = TRUE
  - DieTempCompSrc = TRUE

  NOTE These settings are not applicable for newer PM8150 families.
- Dependency – None

4.7.14 EmergencyShutdownVbatt

This parameter configures the device emergency shutdown limit. The QTI charger app monitors charge current (less than 0 mA) and battery voltage (less than 3.2) for three consecutive reads before initiating emergency shutdown to safeguard the battery. The default value is 3200 mV.

4.7.15 EnableChargerWdog

This parameter enables the charger watchdog to safeguard unintentional charging if the software gets stuck. Charging becomes disabled if the watchdog is configured. Software must pet the watchdog based on its set expiration limit. Supported values are:

- 0 – Do not enable the charger watchdog
- 1 (Default) – Enable the charger watchdog during charging and disable before exiting
- 2 – Enable the charger watchdog during charging and leave enabled when exiting

4.7.16 VBtEmpty threshold

This parameter configures the low battery voltage threshold for the SoC empty interrupt. The default value is 2800 mV.

```
VBtEmpty = 2800
```

4.7.17 VBattEstDiffThreshold

This parameter configures the estimated voltage difference threshold to restart the FG if the threshold difference is higher.

- Default value – 30 mV (Vbatt_Estimate_diff_threshold)
- Dependency – FgCondRestart

Qualcomm
2018-12-26 ..:59:17 PST
sw@wingtech.com

4.7.18 RConnComp compensation resistance

This parameter configures the Rconn compensation resistance.

- Value – In mOhms - range is ±100 mOhms

  NOTE PM8150 families allow up to ±1000 mOhms

- Default value – RconnComp = 0
- Dependency – None

4.8 Battery error handling configuration

The following sections describe battery error handling configuration parameters. See QTI charger app configuration file for LA (for PM8998) for a QTI charger app LA example file.

4.8.1 DebugBoardBatteryIdMin and DebugBoardBatteryIdMax

This parameter specifies the debug board battery ID range. If the voltage associated with the battery ID falls below the range, the QTI charger app handles the battery as a debug board based on Vbatt > boot threshold (3.6 V) allow boot to HLOS; otherwise, shut down.

- Default value – 2000 to 14000 Ohms
- Dependency – DebugBoardBehavior
- See also BootToHLOSThresholdInMv.

4.8.2 SmartBatteryIdMin and SmartBatteryIdMax

This parameter specifies the smart battery ID range. The QTI charger app handles the battery as a smart battery based on Vbatt > 3.6 V allow boot to HLOS; otherwise, continue charging until the configured threshold is reached.

- Default value – 240000 to 450000 Ohms (smart battery ID range)
- See also BootToHLOSThresholdInMv.

4.8.3 RegularBatteryIdMin and RegularBatteryIdMax

This parameter specifies the regular battery ID range. The QTI charger app handles the battery as a regular board based on Vbatt > 3.6 V allow boot to HLOS; otherwise, continue charging until the configured threshold is reached. Default value is 15000 to 137000 Ohms (regular battery ID range).

4.8.4 UnknownBatteryBehavior

This parameter defines unknown battery behavior. It detects if the battery ID is within the specified range. Supported values are:

- 0 – Shut down the device
- 1 – Boot to HLOS if battery is more than threshold, else shut down

Qualcomm
2018-12-20 17:59:17 PST
zk_sw@wingtech.com

- 2 – Conservative charging
- 3 (Default) – Regular charging

4.8.5 DebugBoardBehavior

This parameter defines debug board battery behavior. Supported values are:

- 0 – Show low battery icon, disable PON1/USBIN trigger to prevent reboot and shutdown
- 1 (Default) – Show low battery icon and stay on until device is turned off by user
- 2 – Boot to HLOS

4.8.6 BattMissingCfg

This parameter configures the battery missing detection behavior. Supported values are:

- 0 (Default) – Use battery ID
- 1 – Use battery thermistor
- 2 – Use battery thermistor and ID

4.9 Jeita configuration

The following sections describe Jeita configuration parameters. See QTI charger app configuration file for LA (for PM8998) for a QTI charger app LA example file.

4.9.1 Jeita zones

These parameters enable specified Jeita zones. For negative values, use the negative symbol, for example, -30.

Parameter	Default value (°C)
`JeitaCriticalTempLowLimit`	-20
`JeitaHardColdLimit`	0
`JeitaSoftColdLimit`	10
`JeitaSoftHotLimit`	45
`JeitaHardHotLimit`	60
`JeitaCriticalTempHighLimit`	70

PM8150 default data settings

Parameter	Default value (°C)
`JeitaCriticalTempLowLimit`	-20
`JeitaHardColdLimit`	0
`JeitaSoftColdLimit`	10
`JeitaSoftHotLimit`	40

Qualcomm 2018-12-26 17:59:17 PST k_sw@wingtech.com

Parameter	Default value (°C)
`JeitaHardHotLimit`	45
`JeitaCriticalTempHighLimit`	60

4.9.2 Hardware ADF hot and cold limits

The following parameters configure battery temperature limits in degrees Celcius for PBS temperature monitoring if software is stuck. When the battery temperature goes out of bound, PBS triggers AFP (a shutdown device).

By default, setting the cold limit to -273 disables cold shutdown.

- HwAfpHotLimit = 71
- HwAfpColdLimit = -273

4.9.3 JeitaCcCompCfg

This parameter enables Jeita charge current compensation configuration when the device is within the battery temperature soft limit for the hardware Jeita.

- Jeita compensation values
  - Minimum value – 0 mA
  - Maximum value – 1575 mA
  - Step size – 25 mA
- Default value – 1000 mA

For newer PM8150 families, the hot and cold values can be configured independently.

- JeitaCcCompHotCfg = 1000
- JeitaCcCompColdCfg = 1000

4.9.4 JeitaFvCompCfg

This parameter enables Jeita charge voltage compensation configuration when the device is in battery temperature soft limit for hardware Jeita.

- Jeita compensation values
  - Minimum value – 0 mV
  - Maximum value – 472.5 mV
  - Step size – 7.5 mV
- Default value – 105 mV

For newer PM8150 familes, the hot and cold values can be configured independently.

- JeitaFvCompHotCfg = 105
- JeitaFvCompColdCfg = 105

Qualcomm
2018-12-26 17:59:17 PST
zk_zw@wingtech.com

The Jeita hard limit can be enabled/disabled.

- JeitaHardLimitEnable = TRUE

4.9.5 NoChargeAndWait

This parameter configures device behavior for temperatures outside of the charging range but within the operating range. Supported values are:

- TRUE (Default) – Disable charging and wait
- FALSE – Shut down the device if the temperature is outside of the charging range

4.10 WiPower configuration

The following sections describe WiPower configuration parameters. See QTI charger app configuration file for LA (for PM8998) for a QTI charger app LA example file.

NOTE WiPower is not planned for the PM8150 family, thus this option is not applicable.

4.10.1 WiPowerSupported

This parameter enables WiPower support configuration for the QTI charger app charging device. The default value is TRUE.

4.10.2 DCInBootToHLOSThresholdInMv

This parameter enables WiPower threshold configuration for QTI charger app charging. The default value is 3600 mV.

NOTE WiPower is not planned for the PM8150 family, thus this option is not applicable.

4.10.3 SuspendDCIn

This parameter enables suspended DCIN behavior configuration when WiPower is enabled in QTI charger app charging. The default value is FALSE.

NOTE WiPower is not planned for the PM8150 family, thus this option is not applicable.

4.11 Thermal configuration

The following sections describe thermal configuration parameters. See QTI charger app configuration file for LA (for PM8998) for a QTI charger app LA example file.

4.11.1 SWThermalMitigationEnable

This parameter enables thermal safety mitigation configuration in the QTI charger app charging device. Mitigation is based on the MSM or SDM Tsens maximum average temperature reading. The default value is FALSE.

Qualcomm

4.11.2 TsensTimeoutMins

This parameter enables thermal safety timer configuration when the device is in the thermal zone and not in charger wait state. The device waits for the configured wait time and initiates shutdown if thermal conditions do not normalize.

- Default value – 30 min
- Give-up time in thermal wait for battery disconnect – Max 60 min

4.11.3 Tsens limits or zone

These parameters enable configuration of thermal safety zones or limits. If the temperature is above the extreme temperature limit, the device performs automatic fault protection (AFP).

If the Tsens temperature limit is within the specified wait range (for example, 75 to 90 min), the QTI charger app waits 30 min (polling every 3 sec) to allow the device to cool down. The device performs AFP after the 30 min expiration.

Parameter	Default value (°C)	Description
`TsensHighTemp`	85	High temperature limit for thermal wait
`TsensExtremeTemp`	90	High temperature limit for battery and device safety on battery disconnect
`TsensLowTemp`	75	Low temperature limit for end of thermal wait

Qualcomm

4.12 QTI charger app configuration file for LA (for PM8998)

```
#
# Default Charger App Config settings
#
[CHARGER Config]
#
# Version/Information:
# file ChargerApp_VbattTh_SDM845.cfg
#
# Implements the Qualcomm's Charger application config parameters
#
# Copyright (c) 2016,  Qualcomm Technologies Inc. All rights reserved.
#
#       1 : Initial revision
#       2 : Deleting not needed config params and removing dummy battery2
support
#       3:  Adding Jeita Compensation params
#       4 : Adding parameters for different battery types and QC 3.0 and QC
2.0 chargers
#       5 : Added parameter to support enabling watchdog when charging is
enabled
#       6 : Adding parameters for Aux Coffes, SkinHot and Charger Hot settings
#      7 : Update for Battery profile load
#      8 : Added SupportHostMode
#      9 : Adding Thermal configs
#     10 : Adding support for Charger Fg Peripheral dumps
#     11 : Adding HVDCP Enable control
#     12 : Adding WIPOWER configs
#     13 : Removed config item for setting IUSB_MAX in case of SDP
#     14 : Adding Restarting FG flag
#     15 : Adding Charger led indication config, raising skin hot to 70-80C,
disabling watchdog as default
#     16 : Added changes for supporting different platforms, MTP, QRD, etc.
CfgVersion = 17
--- Threshold Charging Configurations --
#Use Battery SOC or voltage as threshold charging criteria
#Voltage is default
SocOrVoltageBaseBoot = FALSE
#Load Fuel Gauge Battery Profile profile for SOC estimation and accuracy
LoadBatteryProfile   = FALSE
#Below VBAT threshold is used to decide on showing sign of life first before
FG Module Initialization and continuing with threshold charging
DispSignOfLifeMaxThresholdMv = 3700
# FG Conditional Restart on Device reset
FgCondRestart = TRUE
# Charging status indication via led
# 0 = Disable 1 = solid during charging 2 = led blinks during charging
# if turned on LED will be turned off after threhsold charging is completed
i.e. when device boot to HLOS
ChargerLedConfig = 1
--- Threshold Charging Configurations End --
--- Debug Configurations --
# Print Charger DEBUG Messages
PrintChargerAppDbgMsg = FALSE
#Print Charger DEBUG Messages to ULOG File..Default is false
PrintChargerAppDbgMsgToFile = FALSE
#Enable/disable Charger/FG Dump support
EnableChargerFgDump = FALSE
--- ----Debug FG Configurations ------
#dump SRAM contents Default value - FALSE
DumpSram = FALSE
#dump SRAM Start and End Address in Hex Format
#SRAM Block    SRAM Address
#System        0x00 - 0x17
#Profile       0x18 - 0x3C
#Scratchpad    0x50 - 0x7C
#values in decimal
DumpSramStartAddr = 0
#values in decimal
```

Qualcomm
2018-12-26 17:59:17 PST
zk_wewingtech.com

```
DumpSramEndAddr   = 124
#dump SRAM contents timer Duration in s
DumpSramDuration = 90
--- ---- Debug FG Configurations End -----
--- Debug Configurations End --
--- Battery Error Handling Configurations  --
#Battery ID Tolerance Percentage 8%
BatteryIdTolerance = 8
#Debug board ID range, value in Ohms
DebugBoardBatteryIdMin = 0
DebugBoardBatteryIdMax = 14000
#Regular battery ID range, value in Ohms
RegularBatteryIdMin = 15000
RegularBatteryIdMax = 137000
#Smart battery ID range, value in Ohms
SmartBatteryIdMin = 240000
SmartBatteryIdMax = 450000
#Support unknown battery charging behavior
# 0: Shuts down device,    1: Boot to HLOS if battery more than threshold
else shutdown
# 2: Conservative Charging 3: Regular charging
UnknownBatteryBehavior = 3
#Debug board behavior
# 0: Show low battery icon, disable PON1/USBIN trigger to prevent reboot and
shutdown
# 1: Show low battery icon and stay on until device is turned off by user.
# 2: Boot to HLOS
DebugBoardBehavior = 2
#Battery missing config
# 0 = using batt id 1 = using batt therm 2 = both
BattMissingCfg = 0
--- Battery Error Handling Configurations End --
--- Jeita Configurations --
# Configure limits for Battery Temperature (For negative values, use negative
sign. Ex: -30)
JeitaCriticalTempLowLimit = -20
JeitaHardColdLimit = 0
JeitaSoftColdLimit = 10
JeitaSoftHotLimit  = 45
JeitaHardHotLimit  = 60
JeitaCriticalTempHighLimit = 70
#JEITA Charge Current Compensation when in battery temperature soft-limit
#JEITA CC = min is 0 ma and max is 1575 ma - step size is 25mA
JeitaCcCompCfg = 1000
#JEITA Float Voltage Compensation when in battery temperature soft-limit
#min is 0 and max .4725 V step size is 7.5 mV - unit is in mV
JeitaFvCompCfg = 105
```

Qualcomm
2018-12-16 17:59:17 PST
zk_swe@wingtech.com

```
#device behaviour if temp is outside charging range but within operational
range
# 1= Disable charging and wait. 0 = Shutdown device is temp outside
NoChargeAndWait =  TRUE
--- Jeita Configurations End --
--- Initial Configurations  --
#Boot device to HLOS in case of unsupported battery or battery emulator. In
millivolt*/
BootToHLOSThresholdInMv = 3600
#Minimum SOC Threshold before allowing to boot to HLOS
#below param is considered only when SocOrVoltageBaseBoot = TRUE and
LoadBatteryProfile = TRUE
OsStandardBootSocThreshold = 7
# Configure Battery Voltage and Current limit
BattVoltLimHighDelta = 30
# Configure VddMax and IbatMax values
# Set to 0 to configure through API
ChgFvMax  = 4350
ChgFccMax = 2000
#Charging termination current in milliamps
ChargingTermCurrent = 200
# Voltage (in mV) to be reduced from FV_MAX during conservative charging
ConservChgFvDelta  = 200
#Program THERM coeffs ..
#Picked up as per ThermB value per device/battery and initial values are
given in HALF encoded
ProgramAuxThermCoeffs = TRUE
AuxThermC1           = A0
AuxThermC2           = 4F
AuxThermC3           = CF
#based on ThermB and pull up resistor value
AuxThermHalfRangeInC = 25
#Program device Skin and Charger Hot thresholds
ProgramSkinAndChargerHotThreshold = TRUE
DeviceSkinHotInC    = 50
DeviceSkinTooHotInC = 60
ChargerHotInC       = 80
ChargerTooHotInC    = 90
#Lowest Voltage at which device should shutdown gracefully
#value in mV
EmergencyShutdownVbatt = 3200
#Charger WDOG Support options
# 0: Do not enable Charger WDOG
# 1: Enable Charger WDOG during charging and Disable before exiting
# 2: Enable Charger WDOG during charging and leave enabled when exiting
EnableChargerWdog = 1
#Vbat Empty threshold in mv
```

Qualcomm
2018-12-26 17:59:17 PST
zk_xw@wingtech.com

```
VBtEmpty = 2800
--- Initial Configurations  End --
--- Thermal Configurations --
#Enable SW thermal mitigation during charging by default FALSE
# Mitigation is based on MSM Tsens max avg temp reading
SWThermalMitigationEnable = FALSE
## TSENS ##
#High Temperature limit for thermal wait
TsensHighTemp = 85
#High Temperature limit for battey and device safety (battery disconnect)
TsensExtremeTemp = 90
#Low Temperature limit for end of thermal wait
TsensLowTemp = 75
# Give up time in thermal wait for battery disconnect - support up to 60min
TsensTimeoutMins = 30
--- Thermal Configurations End --
--- WiPower Configurations  --
#support wipower or not
WiPowerSupported = FALSE
#Boot device to HLOS in case of wipower charging. In millivolt
DCInBootToHLOSThresholdInMv = 3600
#suspend DCIn or not after exiting UEFI
SuspendDCIn = TRUE
--- WiPower Configurations End --
#
#  End of config
#  Blank line needed after the last config
#
```

Qualcomm
2018-12-26 17:59:17 PST
zk_sw@wingtech.com

4.13 QTI charger app configuration file for LA (for PM8150)

```
# Default Charger App Config settings
#
# CHARGER Config file

#
# Version/Information:
# file QcomChargerConfig_VbattTh_SDM855.cfg
#
# Implements the Qualcomm's Charger application config parameters
#
# Copyright (c) 2017-2018,  Qualcomm Technologies Inc. All rights reserved.
#
#      1 : Initial revision
#      2 : Deleting not needed config params and removing dummy battery2
support
#      3:  Adding Jeita Compensation params
#       4 : Adding parameters for different battery types and QC 3.0 and QC
2.0 charger adapters
#       5 : Added parameter to support enabling watchdog when charging is
enabled
#       6 : Adding parameters for Aux Coffes, SkinHot and Charger Hot settings
#       7 : Update for Battery profile load
#       8 : Added SupportHostMode
#       9 : Adding Thermal configs
#      10 : Adding support for Charger Fg Peripheral dumps
#      11 : Adding HVDCP Enable control
#      12 : Adding WIPOWER configs
#      13 : Removed config item for setting IUSB_MAX in case of SDP
#      14 : Adding Restarting FG flag
#      15 : Adding Charger led indication config, rasing skin hot to 70-80C,
disabling watchdog as default
#      16 : Added changes for supporting different platforms, MTP, QRD, etc.
#      17 : Removed SupportHostMode config (hostmode settings will be done in
USB driver) and other unused config parameters
#      18 : Enable battery profile loading by default
#      19 : Added VBattEstDiffThreshold configuration
#      20 : Add aux coffs
#      21 : Removed  HVDCP Enable control config parameter
#      22 : Adding Config VBatEmpty threshold
#      23 : Adding PBS triggered APF temperature threshold
#      24 : Adding Rconn configuration
#      25 : Adding thermal source configuration to enable/disable INOV
#      26 : Updated config file to platform based sections.
#      27 : Adding debug level mask for file logging
#      28 : ABL min battery voltage config
#      29 : Adding ship mode for factor use case
#      30 : Adding delay before enable batt missing detection
#      31 : Adding Platform HW configuration Override
#
[PlatformCommon]

CfgVersion = 31

#Override default HW configuration assignment
# 0 - None
# 1 - PMIC
#PlatHWCfgOverrdie = 0


#Enable/Disable Ship Mode - when enabled durign factory mode if persistant
ShipMode variable is set then perform shipmode
EnShipMode = FALSE
```

Qualcomm
2018-12-26 17:59:17 PST
zk_swe@wingtech.com

```
#shipmode string to query variable status
ShipModeVarStr = ShipMode

#Minimum Battery Voltage to allow SW Flash Image
SWFlashMinBattVoltageMv = 3600

# Configure Battery Voltage and Current limit
BattVoltLimHighDelta = 30

#Enable SW thermal mitigation during charging by default FALSE
# Mitigation is based on MSM Tsens max avg temp reading
SWThermalMitigationEnable = FALSE

# Print Charger DEBUG Messages
PrintChargerAppDbgMsg = FALSE

#Print Charger DEBUG Messages to ULOG File..Default is false
PrintChargerAppDbgMsgToFile = FALSE

#Debug level mask (in hex)for file logging.. refer to DebugLib.h
FileLoggingDbgLevelMask = 80000042

#Charging termination current in milliamps
ChargingTermCurrent = 200

#Battery ID Tolerance Percentage 8%
BatteryIdTolerance = 8

#Debug board ID range, value in Ohms
DebugBoardBatteryIdMin = 2000
DebugBoardBatteryIdMax = 14000

#Regular battery ID range, value in Ohms
RegularBatteryIdMin = 15000
RegularBatteryIdMax = 137000

#Smart battery ID range, value in Ohms
SmartBatteryIdMin = 240000
SmartBatteryIdMax = 450000

# Voltage (in mV) to be reduced from FV_MAX during conservative charging
ConservChgFvDelta  = 200

#Support unknown battery charging behavior
# 0: Shuts down device,    1: Boot to HLOS if battery more than threshold
else shutdown
# 2: Conservative Charging 3: Regular charging
```

Qualcomm
2018-12-26 17:59:17 PST
zk_swe@wingtech.com

```
UnknownBatteryBehavior = 3

#Debug board behavior
# 0: Show low battery icon, disable PON1/USBIN trigger to prevent reboot and
shutdown
# 1: Show low battery icon and stay on until device is turned off by user.
# 2: Boot to HLOS
DebugBoardBehavior = 2

#Boot device to HLOS in case of unsupported battery or battery emulator. In
millivolt*/
BootToHLOSThresholdInMv = 3600

#Minimum SOC Threshold before allowing to boot to HLOS
#below param is considered only when SocBasedBoot = TRUE and
LoadBatteryProfile = TRUE
OsStandardBootSocThreshold = 7

#device behaviour if temp is outside charging range but within operational
range
# 1= Disable charging and wait. 0 = Shutdown device is temp outside
NoChargeAndWait =  TRUE

#Lowest Voltage at which device should shutdown gracefully
#value in mV
EmergencyShutdownVbatt = 3200

#Battery missing config
# 0 = using batt id 1 = using batt therm 2 = both
BattMissingCfg = 0

#Charger WDOG Support options
# 0: Do not enable Charger WDOG
# 1: Enable Charger WDOG during charging and Disable before exiting
# 2: Enable Charger WDOG during charging and leave enabled when exiting
EnableChargerWdog = 1

#Program BATT_THERM configs
#BiasWait 0 = 0 ms; 1 = 1 ms; 2 = 4 ms; 3 = 12 ms; 4 = 20 ms; 5 = 40 ms; 6 =
60 ms; 7 = 80 ms
#GndSel   TRUE = thermistor is located on the battery pack, FALSE =
thermistor on the PCB(skin temp)
BattThermBiasWait   = 4
BattThermGndSel     = TRUE

#Program AUX_THERM configs
AuxThermBiasWait   = 4
```

Qualcomm
2018-12-26 17:59:17 PST
zk_swe@wingtech.com

```
AuxThermGndSel     = FALSE

#Program device Skin and Charger Hot threhsolds
ProgramSkinAndChargerHotThreshold = TRUE
DeviceSkinHotInC    = 70
DeviceSkinTooHotInC = 80
ChargerHotInC       = 80
ChargerTooHotInC    = 90

#enable/disable charger therm source
SkinTempSrc      = FALSE
DieTempSrc       = TRUE
DieTempCompSrc   = TRUE

#Use Battery SOC or voltage based threshold charging criteria
#FALSE: Voltage threshold based boot-up; TRUE: SOC threshold based boot-up
SocBasedBoot = FALSE

#Load Fuel Gauge Battery Profile profile for SOC estimation and accuracy
LoadBatteryProfile   = TRUE

#Below VBAT threhsold is used to decide on showing sign of life first before
FG Module Initialization and continuing with threshold charging
DispSignOfLifeMaxThresholdMv = 3700

#dump SRAM contents default value is FALSE
DumpSram = FALSE

#dump SRAM Start and End Address in Hex Format
#SRAM Block    SRAM Address
#System        0x00 - 0x23
#Profile       0x41 - 0x10F
#Scratchpad    0x12C - 0x1DF
#values in decimal
DumpSramStartAddr = 0
#values in decimal
DumpSramEndAddr   = 479

#dump SRAM contents timer Duration in s
DumpSramDuration = 30

## TSENS ##
#High Temperature limit for thermal wait
TsensHighTemp = 85
#High Temperature limit for battey and device safety (battery disconnect)
TsensExtremeTemp = 90
#Low Temperature limit for end of thermal wait
```

Qualcomm 2017-12-26 17:59:17 PST zk_sw@wingtech.com

```
TsensLowTemp = 75
# Give up time in thermal wait for battery disconnect - support up to 60min
TsensTimeoutMins = 90

#Enable/disable Charger/FG Dump support
EnableChargerFgDump = FALSE

#Boot device to HLOS in case of DCIn Charging
DCInBootToHLOSThresholdInMv = 3800

#suspend DCIn or not after exiting UEFI
SuspendDCIn = FALSE

# FG Conditional Restart on Device reset
FgCondRestart = TRUE

# VBat Threshold value for conditional restart check, value in mV
VBattEstDiffThreshold = 30

# Charging status indication via led
# 0 = Disable 1 = solid during charging 2 = led blinks during charging
# if turned on LED will be turned off after threhsold charging is completed
i.e. when device boot to HLOS
ChargerLedConfig = 1

#Rconn compensation Resistance, value in mOhms - range is +/- 1000 mOhms
RconnComp = 0

[PlatformMTP]
## Parameter values with respect to MTP Platform (Platform 1)
# Configure FVMax and Fcc values
ChgFvMax  = 4350
ChgFccMax = 2000


# Configure limits for Battery Temperature (For negative values, use negative
sign. Ex: -30)
JeitaCriticalTempLowLimit = -20
JeitaHardColdLimit = 0
JeitaSoftColdLimit = 10
JeitaSoftHotLimit  = 40
JeitaHardHotLimit  = 45
JeitaCriticalTempHighLimit = 60

# Configure battery temperature limits in deg Cfor PBS temperature monitoring
if SW is stuck,
# when battery temperature goes out of bound, pbs will trigger AFP (shutdown
```

Qualcomm
2017-12-26 17:59:17 PST
zk_swewingtech.com

```
device).
# setting cold to -273 to disablecold shutdown
HwAfpHotLimit     = 62
HwAfpColdLimit    = -273

#JEITA Charge Current Compensation when in battery temperature soft-limit
#JEITA CC = min is 0 ma and max is 1575 ma - stpe size is 25mA
JeitaCcCompHotCfg = 1000
JeitaCcCompColdCfg = 1000

#JEITA Float Voltage Compensation when in battery temperature soft-limit
#min is 0 and max .4725 V step size is 7.5 mV - unit is in mV
JeitaFvCompHotCfg = 105
JeitaFvCompColdCfg = 105

#JEITA Hard Limit
JeitaHardLimitEnable = TRUE

#JEITA Float Voltage compensation during soft cold
JeitaSoftColdFvCompEnable = FALSE

#JEITA Float Voltage compensation during soft hot
JeitaSoftHotFvCompEnable = TRUE

#JEITA Charge Current compensation during soft cold
JeitaSoftColdCcCompEnable = TRUE

#JEITA Charge Current compensation during soft hot
JeitaSoftHotCcCompEnable = TRUE

#Vbat Empty threshold in mv
VBatEmpty = 2850

#BATT missing delay in msec
BattMissingDelay = 0
```

Qualcomm
2018-12-26 17:59:17 PST
zk_sweiwingtech.com

MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

A References

A.1 Related documents

Title	Number
Qualcomm Technologies, Inc.	
Understanding PMI8998 Fuel Gauge	80-VT310-138

A.2 Acronyms and terms

Acronym or term	Definition
AFP	Automatic fault protection
BDS	Boot device selection
FG	Fuel gauge
FV	Float voltage
GUID	Globally unique identifier; 128-bit number used to identify an entity (drivers, protocols, files, etc.) within UEFI
INF	Make file for a module that specifies the sources and dependencies
INOV	Intelligent negotiation for optimum voltage
Module	Separate compile-able code or prebuilt library consisting of INF and source code (or binary), such as drivers and libraries
SoC	State of charge
UEFI	Unified extensible firmware interface

Qualcomm
2018-12-26 17:59:17 PST
sw@wingtech.com