Qualcomm Technologies, Inc.

MSM8952.LA Charger Software

User Guide

80-NV610-44 A

April 24, 2015

Confidential and Proprietary – Qualcomm Technologies, Inc.

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

Revision history

Revision	Date	Description
A	Apr 2015	Initial release

Contents

Figures

Tables

1 Introduction

1.1 Purpose

This document describes the programmable features for the charger and the software driver configuration for PMI8952 devices. To properly configure the charging parameters in the device tree file, users must have a complete understanding of the hardware specifications.

1.2 Conventions

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

1.3 Technical assistance

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

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

2 Charger programmable features

2.1 Overview

PMI8952 provides three major functions to the end system:

- Input selection and arbitration
- System output supply and control
- Battery charging

The device is fully programmable via the SPMI interface and configuration is accessible through registers. The switching architecture in conjunction with programmability enables faster charging from current limited inputs such as USB.

2.2 Charge cycle

See Section 3.2 for the programming procedure for the charge cycle.

PMI8952 provides four main charging phases:

- Trickle-charge
- Preconditioning (precharge)
- Constant current (fast charge)
- Constant voltage (taper charge)

The order is not critical because they are set by the driver before the charger is enabled. All phases are fully programmable except trickle charge voltage (2.1 V by default).

Programmable parameters are as follows:

- Precharge current
- Fast-charge current
- Termination current
- Float voltage
- Precharge voltage threshold
- Input current limit (USB single path; dual path USB/DC with ext. PMUX)
- Safety timer duration
- Battery thermal limits

2.2.1 Charge conditions

- Valid charger connection
  - Input voltage < OVLO
  - Input voltage > UVLO
  - Input voltage > VBAT+0.1 V
- Trickle-charge
  - VBAT < 2.1 V
  - IBAT 45 mA
- Precharge
  - VBAT from 2.4 V to 3.0 V
  - IBAT from 100 to 250 mA
- Constant current charge
  - IBAT from 300 mA to 3000 mA
- Constant voltage charge
  - VFLOAT from 3.6 V to 4.5 V
- Charger cycle plot – The programmable charging algorithm is shown in Figure 2-1.

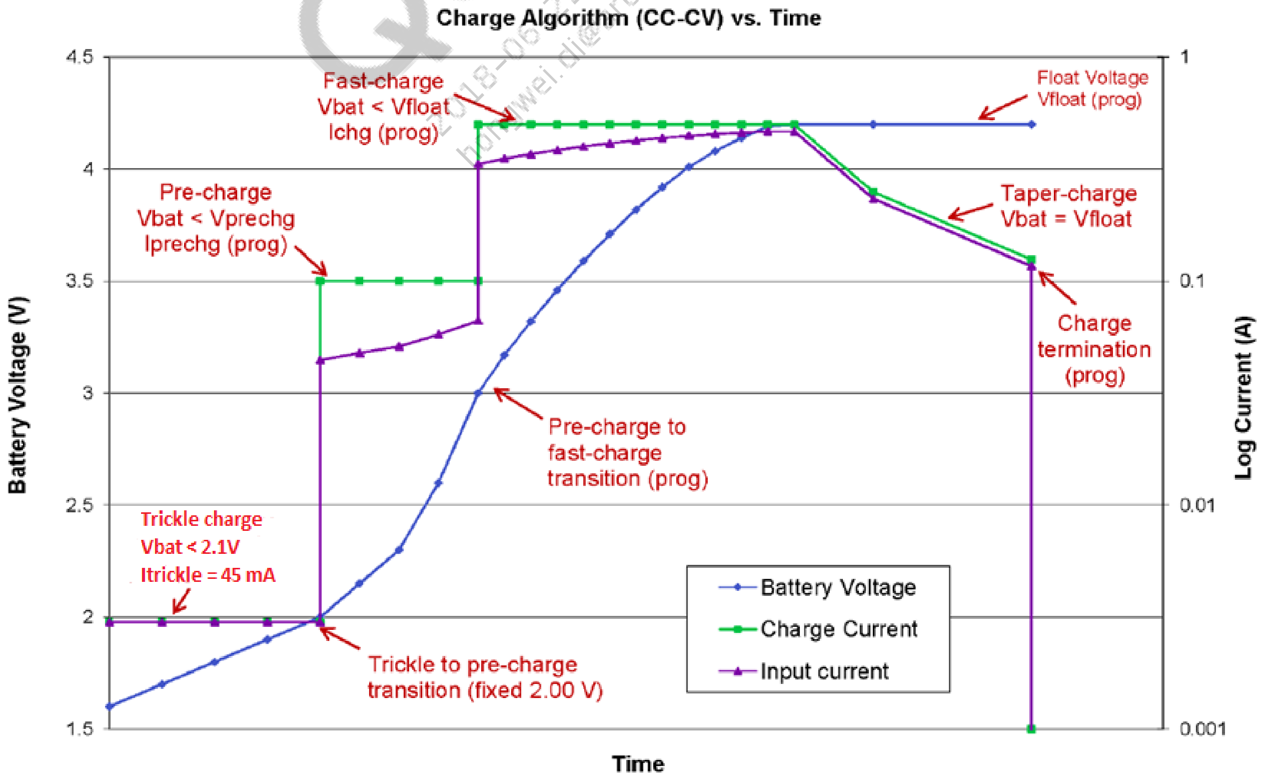


Figure 2-1 Charging algorithm

2.2.2 Charge completion

The charge cycle is considered complete when the charge current reaches the programmed termination current threshold.

- Automatic battery recharge conditions
  - VBAT < VFLOAT - V_RECH
  - Battery temperature returns to normal
- Charger inhibit threshold
  - Programmable to prevent charging initiation upon power cycling or charge enabling or disabling unless the battery voltage is 50 mV, 100 mV, 200 mV, or 300 mV below the float voltage
  - When the charge inhibit function is enabled, the automatic recharge threshold is overridden to the (higher) charge inhibit voltage threshold

2.3 AICL

Automatic Input Current Limit (AICL) is hardware-based. AICL prevents charger voltage collapse by finding the maximum current that the adapter can support and using that as the input current limit. The rule of lowest current wins applies. For example, if AICL is enabled for an adapter capable of supplying 1 A, and the input current limit is programmed for 700 mA, the part limits the input current at 700 mA.

If AICL is enabled, three events can immediately trigger the hardware-based AICL:

- Automatic input current limit operation is running, but not yet completed
- Automatic input current limit operation is completed, but the source voltage has collapsed, that is, AICL did not find the true limit and the system load caused a collapse
- The current setting in the volatile register was updated with a value lower than the AICL setting

Reasons for AICL reruns are as follows:

- AICL only runs once if no rerun function is enabled
- Hardware AICL might stop at a lower current under certain conditions
- ICL is changed to be higher
- PMI8952 provides both USB AICL rerun and DC AICL rerun (if ext. PMUX used for DC)

The hardware periodically reruns AICL after a specific time period if the rerun option is enabled in the beginning. The minimal programmable time of AICL rerun is 45 sec.

To reduce power consumption, AICL is able to rerun only once by enabling the rerun option and then disabling it immediately. It is recommended to initially set the Input Current Limitation (ICL) to the maximum allowable value, and letting AICL adjust it. The ICL is initially set in OTP in the CSIR file. The software can change the ICL for USBIN and DCIN using the following two registers:

- SMBCHGL_USB _USBIN_IL_CFG – 0x000013F2
- SMBCHGL_DC _DCIN_IL_CFG – 0x000014F2

Table 2-1 AICL enable/disable registers

Enable/Disable	Register
USB AICL enable/disable	SMBCHGL_USB_USB_AICL_CFG – 0x000013F3 <bit 2>
DC AICL enable/disable	SMBCHGL_DC_DC_AICL_CFG1 – 0x000014F3 <bit 2>
WiPower™ AICL enable/disable	SMBCHGL_USB_WI_PWR_OPTIONS – 0x000013FF <bit 1>
USB rerun for AICL	SMBCHGL_MISC_CHGR_TRIM_OPTIONS_15_8 – 0x000016F5<bit 5>
DC rerun for AICL	SMBCHGL_MISC_CHGR_TRIM_OPTIONS_15_8 – 0x000016F5<bit 4>

2.4 Battery missing detection

The battery must be present before charging starts. Battery Missing Detection (BMD) can be programmed in four ways:

- BAT_THERM (pin-based)
- BAT_ID (pin-based)
- Battery Missing Algorithm (BMA) – Check at the beginning of the charge cycle (based on the positive poll of the battery)
- BMA – Check every 2.6 sec (based on the positive poll of the battery)
    - For each poll, the system provides a 10 mA discharge current for a short period of time (~100 ms)
    - No additional GPIO pins are required
    - BATT_MISSING_INT can be set by PMI8952 registers
    - The driver can support battery_missing_handler() if necessary
- The default pin source can be either BAT_THERM or BAT_ID, using the qcom,bmd-pin-src configuration parameter:

```
qcom,bmd-pin-src = "bpd_thm_id“
```

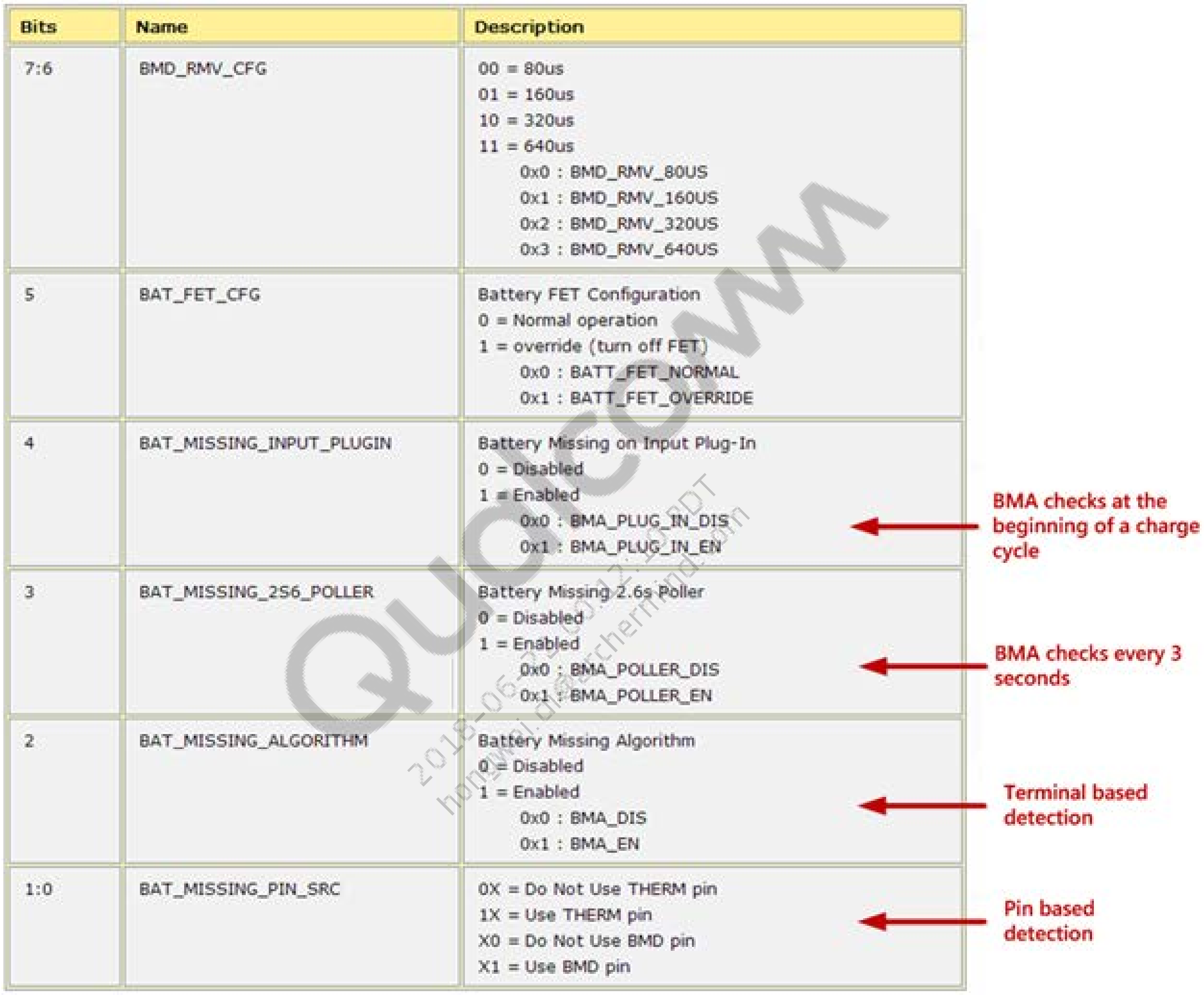
- To enable the battery missing detection algorithm, use the qcom,bmd-algo-disabled configuration parameter:

```
qcom,bmd-algo-disabled=<0>
```

QTI recommends never leaving BAT_ID floating in the hardware design due to the following known issues:

- Setup – Power supply is connected to VBAT; however, BAT_ID is floating and there is no USB connection.
- Issue – Pressing the power key, powers up the device. The device resets in SBL and never boots up.

- Root cause – SYSOK (routed to SHDN) is low because BAT_ID is floating. SBL enables GP1 (same as the external trigger SHDN) reset; after the GP1 S1+S2 timer expires, the device resets.



Bits	Name	Description
7:6	BMD_RMV_CFG	00 = 80us 01 = 160us 10 = 320us 11 = 640us 0x0 : BMD_RMV_80US 0x1 : BMD_RMV_160US 0x2 : BMD_RMV_320US 0x3 : BMD_RMV_640US
5	BAT_FET_CFG	Battery FET Configuration 0 = Normal operation 1 = override (turn off FET) 0x0 : BATT_FET_NORMAL 0x1 : BATT_FET_OVERRIDE
4	BAT_MISSING_INPUT_PLUGIN	Battery Missing on Input Plug-In 0 = Disabled 1 = Enabled 0x0 : BMA_PLUG_IN_DIS 0x1 : BMA_PLUG_IN_EN
3	BAT_MISSING_2S6_POLLER	Battery Missing 2.6s Poller 0 = Disabled 1 = Enabled 0x0 : BMA_POLLER_DIS 0x1 : BMA_POLLER_EN
2	BAT_MISSING_ALGORITHM	Battery Missing Algorithm 0 = Disabled 1 = Enabled 0x0 : BMA_DIS 0x1 : BMA_EN

1:0	BAT_MISSING_PIN_SRC	0X = Do Not Use THERM pin
		1X = Use THERM pin
		X0 = Do Not Use BMD pin
		X1 = Use BMD pin

BMA checks at the beginning of a charge cycle

BMA checks every 3 seconds

Terminal based detection

Pin based detection

Figure 2-2 BMD registers

2.5 JEITA compliance

PMI8952 is compatible with the latest Japan Electronics and Information Technology Industries Association (JEITA) compliance standards. JEITA compliance allows battery charging with reduced charging voltage and/or current outside the conventional battery temperature range as shown in Figure 2-3.

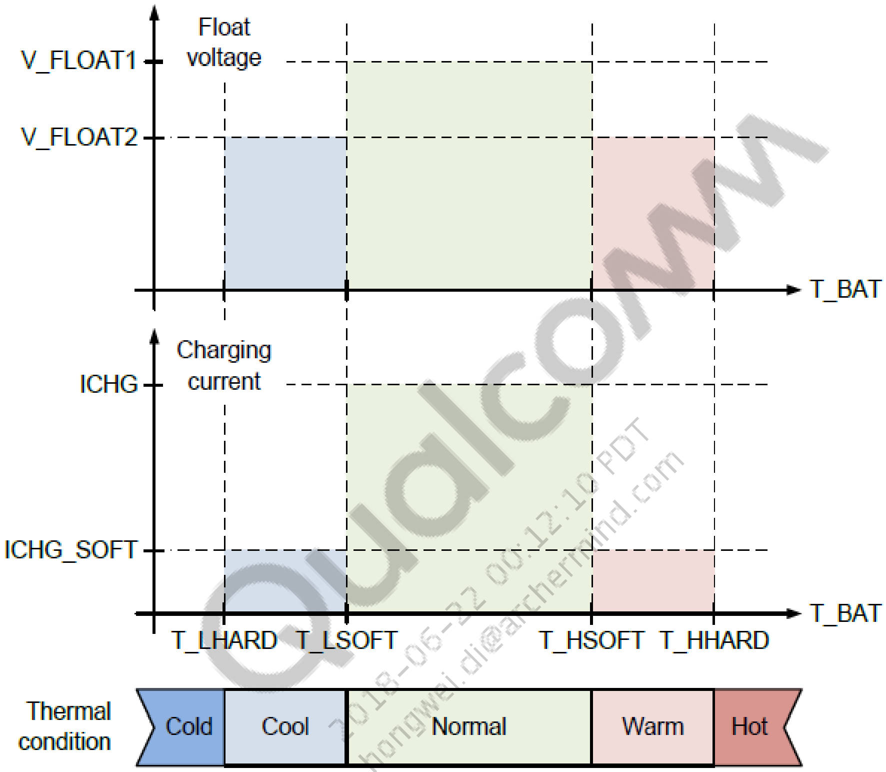


Figure 2-3 Reduced charging voltage and/or current outside conventional battery temperature range

Four thresholds divide the battery temperature into five areas:

- Cold
- Cool
- Normal
- Warm
- Hot

The charging current and float voltage adjust based on thermal zones. This function is hardware-based, but JEITA thresholds can be configured with the software via the device tree. Configuration of soft thresholds (warm and cool) is described in Section 3.2.

2.6 Ship mode

To ensure that the battery does not discharge during shipping for nonremovable battery applications, the PMI8952 can switch the bulk connection of the battery to system FET, reversing the body diode direction and completely disconnecting the battery from the system load.

This mode is entered by setting PMI8952 register 0x1240 bit[0] to 1 after unlocking it (unlock is performed by writing 0xA5 to PMI8952 register 0x12D0). If Ship mode is entered with an input present, the charger buck is disabled first before the body diode direction is changed to prevent excessive current from going to the battery.

Ship mode is exited and normal operation resumed by any of the following methods:

1. Removal of all power resulting in a dVdd_raw_rb
2. Transition of USBIN < ~1 V to USBIN > ~1 V. This is a plug-in on USBIN. This transition is ignored for ~13 sec after triggering Ship mode.
3. Transition of DCIN < ~1 V to DCIN > ~1 V. This is a plug-in on DCIN. This transition is ignored for ~13 sec after triggering Ship mode.
4. KYPDPWR press for at least 10 ms. This transition is ignored for ~13 sec after triggering Ship mode.

For Steps 2, 3, and 4, the ~13 sec Ship mode reset blocking allows time to box the phone without accidentally activating the Ship mode reset.

To set Ship mode from an ADB shell, type the following commands:

```
adb shell
cd /sys/kernel/debug/spmi/spmi-0
echo 0x12D0 > address
echo 0xA5 > data
echo 0x1240 > address
echo 0x01 > data
```

3 Charger driver configuration

3.1 PMI8952 charger driver

Source code for the charger driver software is at kernel//drivers/power/qpnp-smbcharger.c.

For charger device configuration, see DTS at kernel/arch/arm64/boot/dts/qcom/msm-pmi8950.dtsi.

DTSI documentation is located at kernel/Documentation/devicetree/bindings/power/qpnp-smbcharger.txt.

3.2 Configuring the software

The configuration parameters described in Table 3-1 are used to customize the charger driver through the device tree file. The definition of charger parameters can also be found in kernel/Documentation/devicetree/bindings/power/qpnp-smbcharger.txt.

Table 3-1 Configuration parameters

Configuration parameter	Description
Required properties	
qcom,chgr	Supports charging control and status reporting.
qcom,bat-if	Battery status reporting such as presence, temperature reporting, and voltage collapse protection.
qcom,usb-chgpth	USB charge path detection and input current limiting configuration.
qcom,dc-chgpth	DC charge path detection and input current limiting configuration.
qcom,chg-misc	Miscellaneous features such as safety timers and SYSOK pin control.
qcom,chg-otg	OTG configuration control.
compatible	Must be qcom,qpnp-smbcharger.
spmi-dev-container	Must be included in the parent node to set up the SPMI USB node devices.
#address-cells	Must be <1>.
#size-cells	Must be <1>.
reg	The SPMI initial address and its length for this peripheral.
interrupts	Specifies the interrupt associated with the peripheral <slave_id peripheral_id offset>.
interrupt-names	Specifies the interrupt names for the peripheral; every available interrupt must have an associated name with it to identify its purpose.

Configuration parameter	Description
interrupt-names for qcom,chgr	▪ chg-tcc-thr – Triggers on charge completion ▪ chg-taper-thr – Triggers on the taper charge transition ▪ chg-inhibit – Notifies when battery voltage is too high to resume charging ▪ chg-p2f-thr – Triggers on transitioning from precharge to fast charge ▪ chg-rechg-thr – Triggers on battery voltage falling below the resume threshold
interrupt-names for qcom,bat-if	▪ batt-hot – Triggers on battery temperature hitting the hot threshold; charging stops ▪ batt-warm – Triggers on the battery temperature hitting the warm threshold; charging current is reduced ▪ batt-cool – Triggers on the battery temperature hitting the cool threshold; charging current is reduced ▪ batt-cold – Triggers on the battery temperature hitting the cold threshold; charging stops ▪ batt-missing – Battery missing status interrupt ▪ batt-low – Triggers on battery voltage falling across the VBATLOW threshold
interrupt-names for qcom,usb-chgpth	▪ usbin-uv – USB input voltage falls below the valid USBIN threshold (the lower of 3.65 V and VBAT+180 mV) ▪ usbin-src-det – USB automatic power source detection finishes
interrupt-names for qcom,dc-chgpth	▪ dcin-uv – DC input voltage falls below the valid DCIN threshold
interrupt-names for qcom,chgr-misc	▪ safety-timeout-mins – Charger watchdog timer interrupt ▪ temp-shutdown – Triggers when the charger goes over temperature and causes a shutdown ▪ power-ok – Triggers when the charger switcher turns on or off
Optional properties	
qcom,iterm-ma	Specifies the termination current to indicate EoC; possible values in mA are 50, 100, 150, 200, 250, 300, 500, and 600.
qcom,float-voltage-mv	Float voltage in mV – Maximum voltage up to which the battery is charged; supported range is 3.6 V to 4.5 V.
qcom,float-voltage-comp	Specifies the JEITA float voltage compensation. Value ranges from 0 to 63.
qcom,soft-vfloat-comp-disabled	Set this property when the battery is powered via external source and could go above the float voltage.
qcom,autoadjust-vfloat	A Boolean property that when set, makes the driver automatically readjust vfloat using the fuel gauge ADC readings to make charging more accurate.
qcom,fastchg-current-ma	Specifies the fast charge current in mA. Supported range is from 300 mA to 3000 mA.
qcom,fastchg-current-comp	Specifies the fast charge current compensation in mA. Supported values are 250, 700, 900 and 1200mA.
qcom,resume-delta-mv	Specifies minimum voltage drop in mV below float voltage required to initiate a new charging cycle; supported values are 50, 100, 200, and 300 mV.
qcom,charging-timeout	Maximum duration in minutes that a single charge cycle may last; supported values are 0, 192, 384, 768, and 1536.
qcom,precharging-timeout	Maximum duration in minutes that a single precharge cycle may last; supported values are 0, 24, 48, 96, and 192.

Configuration parameter	Description
qcom,battery-psy-name	Name of the main battery power supply that the charger is to register; failing to define this property defaults the name to “battery”.
qcom,bms-psy-name	Psy name for reporting battery capacity. If left unspecified, the capacity uses a preprogrammed default value of 50.
qcom,dc-psy-type	The type of charger connected to the DC path; can be "Mains" or "Wireless."
qcom,dc-psy-ma	The current in mA that the DC path can support; must be specified if dc-psy-type is specified; valid range is 300 mA to 2000 mA.
qcom,dcin-vadc	The phandle to PMI8952 voltage ADC. The ADC is to get notifications when the DCIN voltage crosses a programmed min/max threshold. This is used to make configurations for optimized power draw for WiPower.
qcom,low-volt-dcin	A Boolean property which upon being set enables the AICL deglitch configuration dynamically. This must be set if the DCIN supply is going to be less than or equal to 5 V.
qcom,wipower-div2-ilim-map	Array of 5 elements to indicate the voltage ranges and their corresponding current limits. The 5 elements with index [0..4] are: ▪ [0] – voltage_low in uV ▪ [1] – voltage_high in uV ▪ [2] – Current limit for pass through in mA ▪ [3] – Current limit for DIV2 mode DCIN low voltage in mA ▪ [4] – Current limit for DIV2 mode DCIN high voltage in mA The DIV2 and PT tables indicate the current limits to use when WiPower is operating in divide_by_2 mode pass-through mode respectively. The default table is used when the voltage ranges beyond the ones specified in the mapping table. If dcin-vadc or any of these mapping tables are not specified, dynamic DCIN input is disabled.
qcom,wipower-pt-ilim-map	
qcom,wipower-default-ilim-map	
qcom,charging-disabled	Set this to disable charging in the build by default
qcom,resume-delta-mv	Specifies the minimum voltage drop in millivolts below the float voltage that is required in order to initiate a new charging cycle; supported values are 50, 100, 200, and 300 mV.
qcom,chg-inhibit-en	Boolean that indicates whether the charge inhibit feature must be enabled. If this is not set, the charge inhibit feature is disabled by default.
qcom,chg-inhibit-fg	Indicates if the recharge threshold source has to be Fuel gauge ADC. If this is not set, the sensor is analog by default.
qcom,bmd-algo-disabled	Indicates if the battery missing detection algorithm is disabled; if this node is present, SMB uses the THERM pin for battery missing detection.
qcom,bmd-pin-src	A string that indicates the source pin for the battery missing detection: ▪ bpd_none – Battery is considered always present ▪ bpd_id – Battery ID pin is used ▪ bpd_thm – Battery therm pin is used ▪ bpd_thm_id – Both pins are used (battery is considered missing if either pin is floating)
qcom,iterm-disabled	Disables the termination current feature; this is a Boolean property.
qcom,ibat-ocp-threshold-ua	Maximum current before the battery triggers overcurrent protection. Use the recommended battery pack value minus some margin.

Configuration parameter	Description
qcom,thermal-mitigation	Array of input current limit values for different system thermal mitigation levels. A flat array that denotes the maximum charge current in mA for each thermal level.
qcom,jeita-temp-hard-limit	Enables or disables the JEITA temperature hard limit based on the value 1 or 0. Specify 0 if the JEITA temperature hard limit must be disabled. If it is not present, JEITA temperature hard limit is based on what the boot loader had set earlier.
qcom,parallel-usb-min-current-ma	Minimum current drawn by the primary charger before enabling the parallel charger if one exists; do not define this property if no parallel chargers exist.
qcom,parallel-usb-9v-min-current-ma	Minimum current drawn by the primary charger before enabling the parallel charger if one exists. This property applies only for 9 V HVDCP chargers.
qcom,parallel-allowed-lowering-ma	Acceptable current drop from the initial limit to keep parallel charger activated. If the charger current reduces beyond this threshold parallel charger is disabled. Must be specified if parallel charger is used.
qcom,battery-data	Points to the phandle of a node which contains the battery-profiles supported by the charger/FG.
qcom,charge-unknown-battery	Boolean that indicates whether an unknown battery without a matching profile is to be charged. If this is not set, and the fuel gauge does not recognize the battery based on its battery ID, the charger does not start charging.
otg-parent-supply	A phandle to an external boost regulator for OTG if it exists.
qcom,rparasitics-uohm	Parasitic resistance of the board following the line from the battery connectors through vph_power. This is used to calculate maximum available current of the battery.
qcom,vled-max-uv	Maximum input voltage of the flash LEDs. This is used to calculate maximum available current of the battery.
qcom,chg-led-support	A bool property to support the charger led feature.
qcom,chg-led-sw-controls	A bool property to allow the software to control the charger led without a valid charger.

3.2.1 Charger device tree example

The following example demonstrates using the device tree to configure the PMI8952 charger.

```
Example:

pmi8950_charger: qcom,qpnp-smbcharger {
   spmi-dev-container;
   compatible = "qcom,qpnp-smbcharger";
   #address-cells = <1>;
   #size-cells = <1>;

   qcom,iterm-ma = <100>;
   qcom,fastchg-current-ma = <2000>;
   qcom,float-voltage-mv = <4200>;
   qcom,resume-delta-mv = <200>;
   qcom,chg-inhibit-fg;
   qcom,rparasitic-uohm = <100000>;
   qcom,bms-psy-name = "bms";
   qcom,thermal-mitigation = <1500 700 600 0>;
   qcom,parallel-usb-min-current-ma = <1400>;
   qcom,parallel-usb-9v-min-current-ma = <900>;
   qcom,parallel-allowed-lowering-ma = <500>;

   qcom,chgr@1000 {
        reg = <0x1000 0x100>;
        interrupts =  <0x2 0x10 0x0>,
                           <0x2 0x10 0x1>,
                           <0x2 0x10 0x2>,
                           <0x2 0x10 0x3>,
                           <0x2 0x10 0x4>,
                           <0x2 0x10 0x5>,
                           <0x2 0x10 0x6>,
                           <0x2 0x10 0x7>;

        interrupt-names =  "chg-error",
                                "chg-inhibit",
                                "chg-prechg-sft",
                                "chg-complete-chg-sft",
                                "chg-p2f-thr",
                                "chg-rechg-thr",
                                "chg-taper-thr",
                                "chg-tcc-thr";
   };
```

```
qcom,otg@1100 {
     reg = <0x1100 0x100>;
     interrupts =  <0x2 0x11 0x0>,
                        <0x2 0x11 0x1>,
                        <0x2 0x11 0x3>;
     interrupt-names =  "otg-fail",
                             "otg-oc",
                             "usbid-change";
};

qcom,bat-if@1200 {
     reg = <0x1200 0x100>;
     interrupts =  <0x2 0x12 0x0>,
                        <0x2 0x12 0x1>,
                        <0x2 0x12 0x2>,
                        <0x2 0x12 0x3>,
                        <0x2 0x12 0x4>,
                        <0x2 0x12 0x5>,
                        <0x2 0x12 0x6>,
                        <0x2 0x12 0x7>;

     interrupt-names =  "batt-hot",
                             "batt-warm",
                             "batt-cold",
                             "batt-cool",
                             "batt-ov",
                             "batt-low",
                             "batt-missing",
                             "batt-term-missing";
};

qcom,usb-chgpth@1300 {
     reg = <0x1300 0x100>;
     interrupts =  <0x2 0x13 0x0>,
                        <0x2 0x13 0x1>,
                        <0x2 0x13 0x2>,
                        <0x2 0x13 0x5>;
     interrupt-names =  "usbin-uv",
                             "usbin-ov",
                             "usbin-src-det",
                             "aicl-done";
};
```

```
    qcom,dc-chgpth@1400 {
         reg = <0x1400 0x100>;
         interrupts =  <0x2 0x14 0x0>,
                            <0x2 0x14 0x1>;
         interrupt-names =  "dcin-uv",
                                 "dcin-ov";
    };

    qcom,chgr-misc@1600 {
         reg = <0x1600 0x100>;
         interrupts =  <0x2 0x16 0x0>,
                            <0x2 0x16 0x1>,
                            <0x2 0x16 0x2>,
                            <0x2 0x16 0x3>,
                            <0x2 0x16 0x4>,
                            <0x2 0x16 0x5>;

         interrupt-names =  "power-ok",
                                 "temp-shutdown",
                                 "safety-timeout",
                                 "flash-fail",
                                 "otst2",
                                 "otst3";
    };
};
```

3.3 End of charge

The charger uses the hardware-based End-of-Charge (EoC), which is generated when IBAT < ITERM. The current termination threshold is configured with the qcom,iterm-ma configuration parameter in the device tree.

3.4 Disable or enable the charger

The charger can be enabled or disabled using either USB or ADB.

Disable or enable the charger via USB

1. Type the following ADB commands:

```
adb root
adb wait-for-devices
adb shell setprop persist.usb.chgdisabled 1
adb root
```

2. To enable the charger, replace 1 with 0.

Disable or enable the charger via Wi-Fi

1. Type the following ADB commands:

```
adb root
adb wait-for-devices
adb shell setprop persist.adb.tcp.port 5555 && adb tcpip
adb connect <the target ip>
adb shell setprop persist.usb.chgdisabled 1
```

2. To enable the charger, replace 1 with 0.

3.5 Charger interrupts

Different fault conditions can initiate an interrupt (IRQ) output. These conditions can be selected via the corresponding register. Interrupts must be registered in the device tree so that they can be automatically mapped at boot. No driver code modifications are necessary.

Table 3-2 Charger interrupts

Interrupt	Description
batt_hot_irq	Battery is in the hot zone (hot hard limit threshold).
batt_warm_irq	Battery is in the warm zone (hot soft limit threshold).
batt_cool_irq	Battery is in the cool zone (cold soft limit threshold).
batt_cold_irq	Battery is in the cold zone (cold hard limit threshold).
batt_missing_irq	Battery missing is detected.
vbat_low_irq	Battery voltage is low (programmable from 2.5 V to 3.58 V).
chg_hot_irq	Called when the die temperature reaches 140°C.
chg_term_irq	End of charge.
taper_irq	Start to taper charge.
recharge_irq	Battery recharges.
safety_timeout_irq	Charger safety timer expires.
power_ok_irq	Called when the switcher turns on or off.
dcin_uv_irq	Called when the DC voltage crosses the UV threshold.
usbin_uv_irq	Called when the USB voltage crosses the UV threshold.
src_detect_irq	Called when USB charger type is detected.
chg_inhibit_irq	Called when charger is inserted and the battery voltage is high.
chg_hot_irq	Used primarily for thermal mitigation; it notifies the driver that the charger is going to shut down (see Figure 3-1).

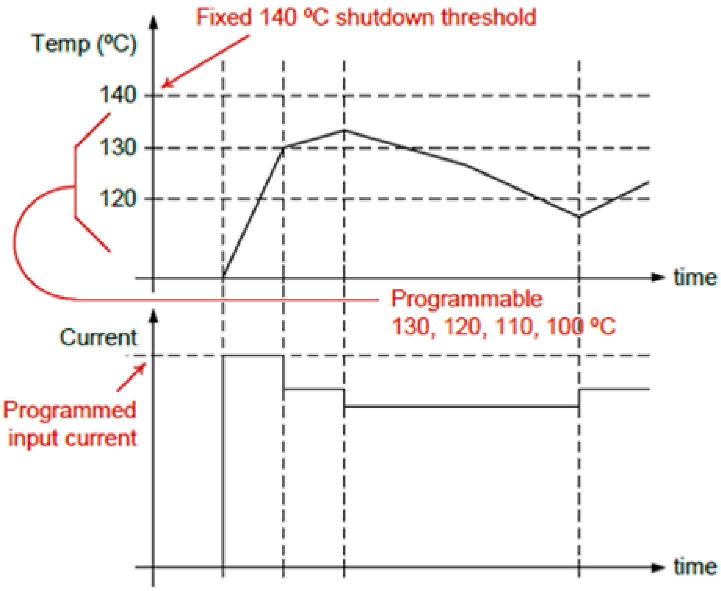


Figure 3-1 Charger shutdown

3.6 PMI8952 parallel charging with SMB1351

Follow the procedures in this section to run two chargers (PMI8952 and SMB1351) simultaneously and enable parallel charging.

Ensure that the smb1351-charger driver is in the software build

1. In /kernel/drivers/power/Kconfig, add the following lines:

```
config SMB1351_USB_CHARGER
tristate "smb1351 usb charger (with VBUS detection)"
depends on I2C
```

2. In /kernel/drivers/power/Makefile, add the following line:

```
obj-$(CONFIG_SMB1351_USB_CHARGER) += smb1351-charger.o
```

3. In arch/arm64/configs/msm-perf_defconfig and arch/arm64/configs/msm_defconfig, add the following line:

```
"CONFIG_SMB1351_USB_CHARGER=y" in the appropriate defconfig file.
```

If only the PMI8952 charger is required, remove the SMB1351 configuration by deleting the above commands.

Modify the device tree

1. Add the following node to the I2C bus node to enable the SMB1351 charger in parallel charging mode.

```
smb1351-charger@1d {
     compatible = "qcom,smb1351-charger";
     reg = <0x1d>;
          qcom,parallel-charger;
          qcom,float-voltage-mv = <4400>;
          qcom,recharge-mv = <100>;
     };
```

2. Add the following property to the qpnp-smbcharger node.

```
qcom,parallel-usb-min-current-ma = <1000>;
```

3.7 PMI8952 charger bringup without battery profile

To allow charging to proceed with an unknown battery profile (for example, for bringup purposes before battery characterization), include the following device tree property in your charger device tree node (`qcom,qpnp-smbcharger`):

```
qcom,charge-unknown-battery;
```

A References

A.1 Acronyms and terms

Acronym or term	Definition
AICL	automatic input current limit
BMA	battery missing algorithm
BMD	battery missing detection
EoC	end of charge
JEITA	Japan Electronics and Information Technology Industries Association
SPMI	system power management interface