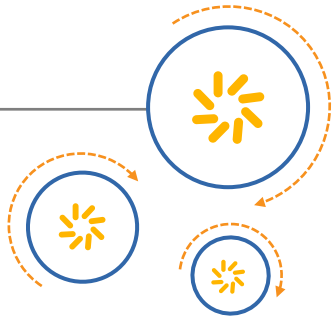
Qualcomm Technologies, Inc.



MSM8994.LA Charger Software

User Guide

80-NM328-56 B

April 14, 2015

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

Revision history

Revision	Date	Description
A	Jul 2014	Initial release
B	Apr 2015	Added Section 2.6, updated Sections 2.4 and 3.7

Contents

Figures

Tables

1 Introduction

1.1 Purpose

This document describes the programmable features for the charger and the software driver configuration for the PMI8994. It is intended for customers using the PMI8994 chip. 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, e.g., `#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

The PMI8994 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.

The PMI8994 provides four main charging phases:

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

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

Programmable parameters are:

- Precharge current
- Fast-charge current
- Termination current
- Float voltage
- Precharge voltage threshold
- Input current limit (DCIN/USBIN)
- 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.

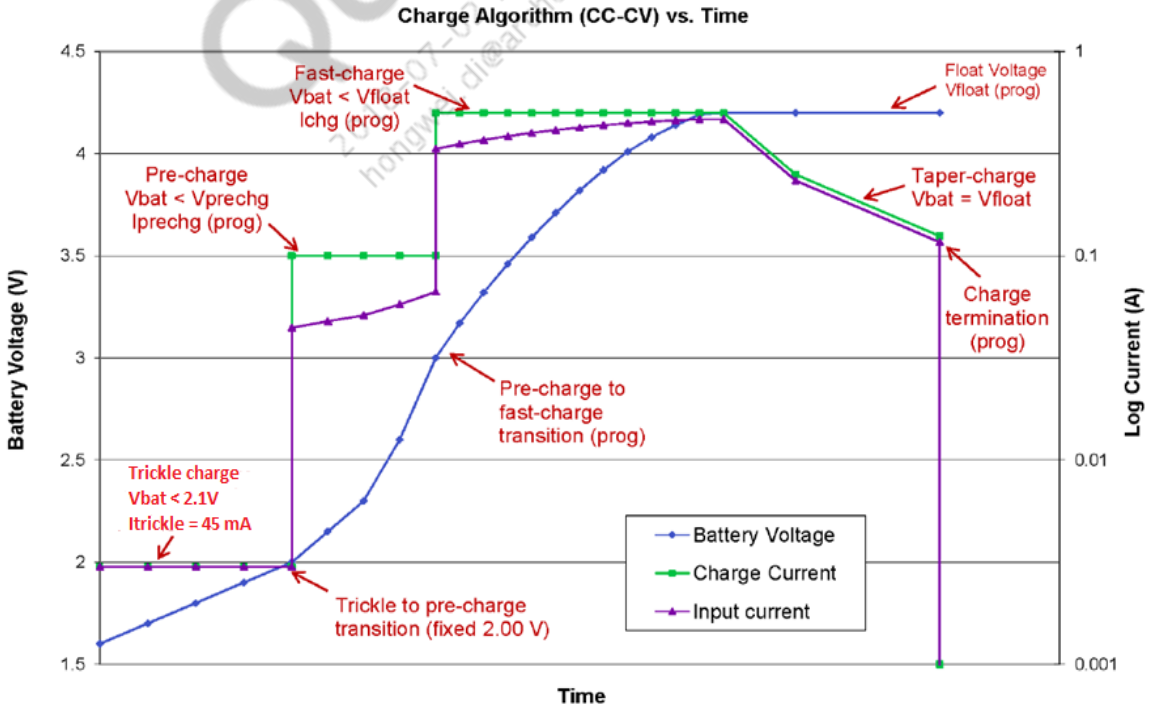


Figure 2-1 Charging algorithm

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
  - This is programmable to prevent charging initiation upon power cycling or charge enabling/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 will be 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; e.g., if AICL is enabled for an adapter capable of supplying 1 A and the input current limit is programmed for 700 mA, the part will limit 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, e.g., 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:

- 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.
- PMI8994 provides both USB AICL rerun and DC AICL rerun.

The hardware will periodically rerun 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 suggested that the Input Current Limitation (ICL) should initially be set to the maximum allowable value and then let 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:

- SMBCHG_USB_CHGPTH_USBIN_IL_CFG – 0x000013F2
- SMBCHG_DC_CHGPTH_DCIN_IL_CFG – 0x000014F2

AICL enable/disable registers are shown in Table 2-1.

Table 2-1 AICL enable/disable registers

Enable/Disable	Register
USB AICL enable/disable	SMBCHG_USB_CHGPTH_USB_AICL_CFG – 0x000013F3 <bit 2>
DC AICL enable/disable	SMBCHG_DC_CHGPTH_DC_AICL_CFG1 – 0x000014F3 <bit 2>
WiPower AICL enable/disable	SMBCHG_USB_CHGPTH_WI_PWR_OPTIONS – 0x000013FF <bit 1>
USB rerun for AICL	SMBCHG_MISC_CHGR_TRIM_OPTIONS_15_8 – 0x000016F5<bit 5>
DC rerun for AICL	SMBCHG_MISC_CHGR_TRIM_OPTIONS_15_8 – 0x000016F5<bit 4>

2.4 Battery Missing Detection (BMD)

The battery must be present before charging starts. 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 needed.
  - BATT_MISSING_INT can be set by PMI8994 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 leavingBAT_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.

The BMD registers are shown in Figure 2-2.

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

PMI8994 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.

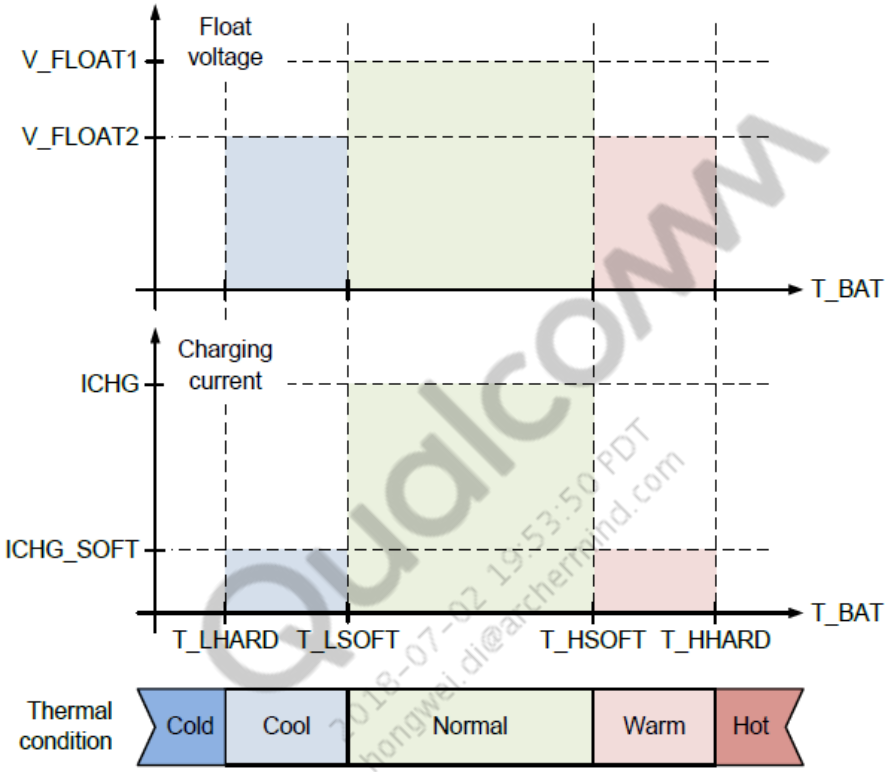


Figure 2-3 Reduced charging voltage and/or current outside conventional battery temperature range

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

NOTE: This section was added to this document revision.

To ensure that the battery does not discharge during shipping for nonremovable battery applications, the PMI8994 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 PMI8994 register 0x1240 bit[0] to 1 after unlocking it (unlock is performed by writing 0xA5 to PMI8994 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, execute 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 PMI8994 charger driver

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

For charger device configuration, see DTS at kernel/arch/arm/boot/dts/qcom/msm-pmi8994.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 needs to 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 a low threshold
interrupt-names for qcom,usb-chgpth	▪ usbin-uv – USB input voltage falls below a valid threshold ▪ usbin-src-det – USB automatic source detection finishes
interrupt-names for qcom,dc-chgpth	▪ dcin-uv – DC input voltage falls below a valid 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,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
qcom,battery-psy-name	The name of the main battery power supply that the charger will register; failing to define this property will default the name to “battery"
qcom,bms-psy-name	The psy name to use 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,charging-disabled	Set this if charging should be disabled in the build by default

Configuration parameter	Description
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,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,thermal-mitigation	Array of input current limit values for different system thermal mitigation levels; should be a flat array that denotates the maximum charge current in mA for each thermal level
qcom,soft-vfloat-comp-disabled	Set this property when the battery is powered via external source and could go above the float voltage
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

Use the device tree to configure the PMI8994 charger. A charger device tree example is shown here.

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

        qcom,iterm-ma = <100>;
        qcom,float-voltage-mv = <4200>;
        qcom,resume-delta-mv = <100>;
        qcom,bmd-pin-src = "bpd_thm_id";
        qcom,dc-psy-type = "Mains";
        qcom,dc-psy-ma = <1500>;
        qcom,bms-psy-name = "bms";
        qcom,battery-psy-name = "battery";
        qcom,thermal-mitigation = <1500 700 600 325>;

        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>;
     };

     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 0x3>,
                          <0x2 0x13 0x4>,
                          <0x2 0x13 0x5>,
                          <0x2 0x13 0x6>;

            interrupt-names =  "usbin-uv",
                               "usbin-ov",
                               "usbin-src-det",
                               "otg-fail",
                               "otg-oc",
                               "aicl-done",
                               "usbid-change";
        };

        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 (EoC)

The charger utilizes the hardware-based 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/enable the charger

Disable/enable the charger via USB as follows:

5. Enter the following ADB commands:
```
adb root
adb wait-for-devices
adb shell setprop persist.usb.chgdisabled 1
adb root
```
6. To enable the charger, replace 1 with 0.

Disable/enable the charger via Wi-Fi as follows:

1. Enter 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 should be registered in the device tree so that they can be automatically mapped at boot. No driver code modifications should be necessary. Interrupts are shown in Table 3-2.

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

Interrupt	Description
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 as shown in Figure 3-1
power_ok_irq	Used primarily for WiPower, it notifies the driver that WiPower transmission is ready as shown in Figure 3-2

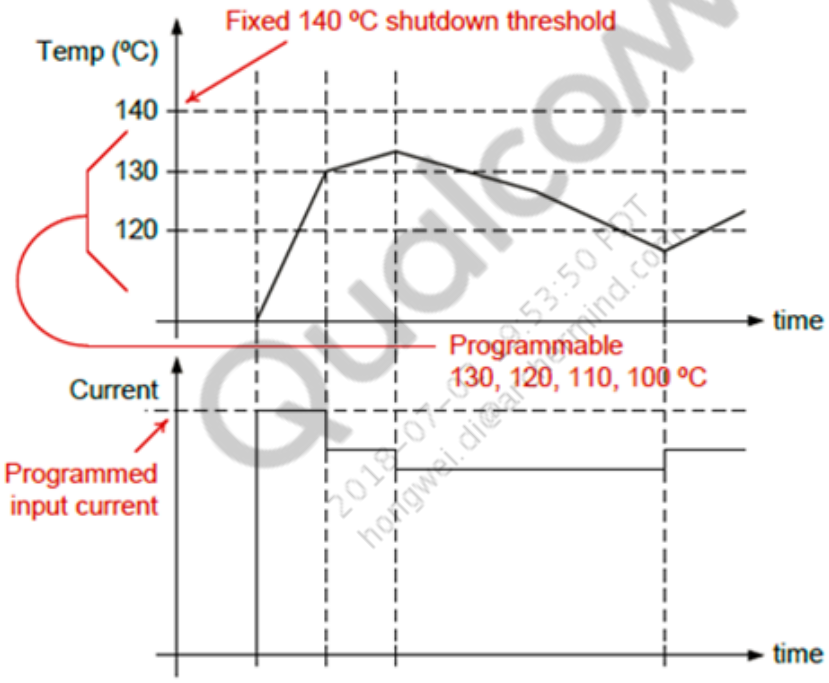
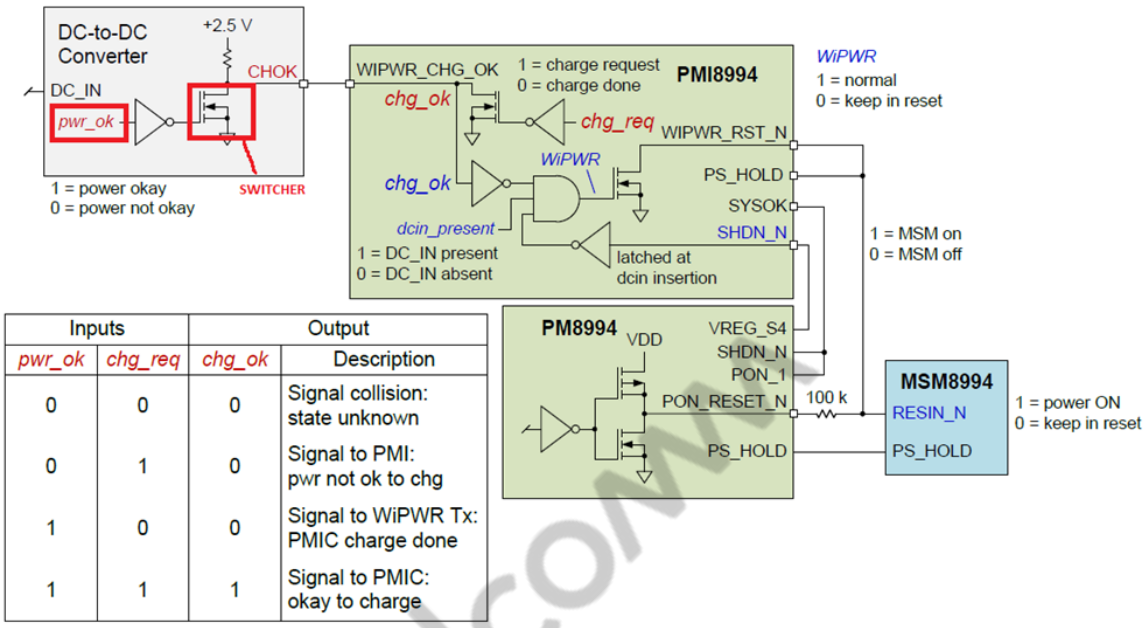


Figure 3-1 Charger shutdown



Inputs		Output	
pwr_ok	*chg_req*	*chg_ok*	Description
0	0	0	Signal collision: state unknown
0	1	0	Signal to PMI: pwr not ok to chg
1	0	0	Signal to WiPWR Tx: PMIC charge done
1	1	1	Signal to PMIC: okay to charge

Figure 3-2 Wipower transmission ready

3.6 PMI8994 parallel charging with SMB1357

To run two chargers (PMI8994 and SMB1357) simultaneously, ensure the smb135x-charger driver is in the software build as follows:

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

```
config SMB135X_CHARGER
tristate "SMB135X Battery Charger"
depends on I2C
```

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

```
obj-$(CONFIG_SMB135X_CHARGER) += smb135x-charger.o
```

3. In arch/arm/configs/msm8994-perf_defconfig and arch/arm/configs/msm8994_defconfig, add the following line:

```
"CONFIG_SMB135X_CHARGER=y" in the appropriate defconfig file.
```

If only the PMI8994 charger is needed, remove the SMB1357 configuration by deleting the above commands.

To enable parallel charging, the following changes are required in the device tree.

1. Add the following node to the I2C bus node to enable the SMB1357 charger in parallel charging mode.

```
smb1357-charger@1c {
     compatible = "qcom,smb1357-charger";
     reg = <0x1c>;
          qcom,parallel-charger;
          qcom,float-voltage-mv = <4250>;
          qcom,recharge-thresh-mv = <100>;
     };
```

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

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

3.7 PMI8994 charger bringup without battery profile

NOTE: Numerous changes were made in this section.

Include the following device tree property in your charger device tree node (`qcom,qpnp-smbcharger`):

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

A References

A.1 Related documents

Title	Number
PM8994 and PMI8994 Power Management ICS Design Guidelines/Training Slides	80-NJ117-5
PM8994 Power Management IC Device Specification	80-NJ117-1
PMI8994 Power Management IC Device Specification	80-NJ118-1
MSM8994.LA Linux PMIC Software Drivers Overview w/Audio	AU80-NM328-40

A.2 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