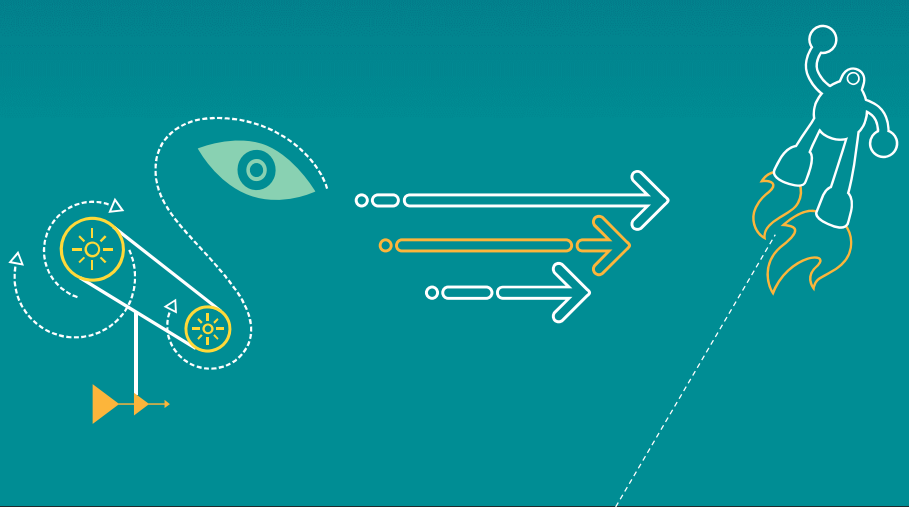
QPNP Charger Driver Details

Qualcomm Technologies, Inc.



80-NG432-1 D

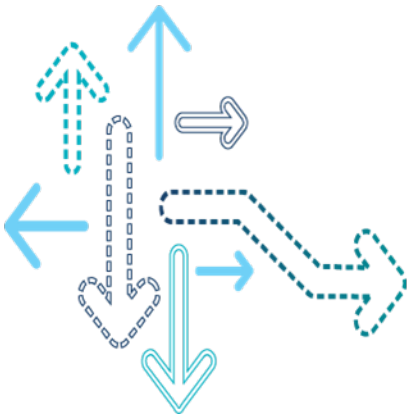
Qualcomm Technologies, Inc.
5775 Morehouse Drive
San Diego, CA 92121
U.S.A.

Revision History

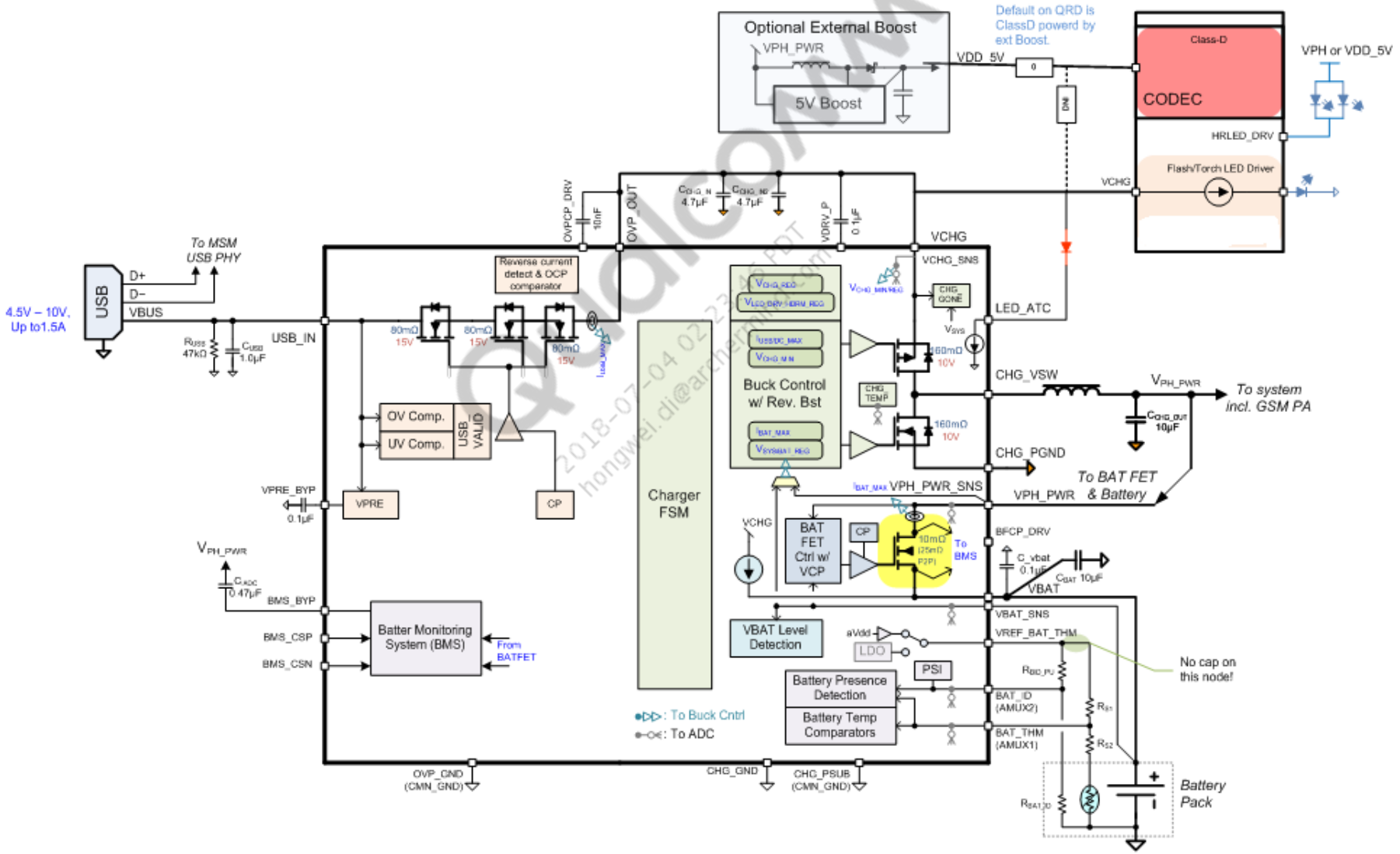
Revision	Date	Description
A	Apr 2013	Initial release
B	May 2013	Updated QPNP charger system, device tree, and driver; added power supply information
C	Nov 2013	Added slides 5-14 and 43-61
D	Mar 2015	Updated slide 38

Contents

QPNP Charger Blocks



QPNP Charger Blocks



QPNP Charger Blocks (cont.)

- Major blocks of the QPNP charger:
  - Charger peripheral (SMBB_CHGR)
  - Buck peripheral (SMBB_BUCK)
  - Battery interface (SMBB_BAT_IF)
  - USB charging path (SMBB_USB_CHGPTH)
  - Switch Mode Battery charger and Boost (SMBB) boost (SMBB_BOOST)
  - SMBB miscellaneous (SMBB_MISC)
  - Coin cell charger (COIN)

SMBB Charger Peripherals

An SMBB charger is divided into seven peripherals:

- SMBB_CHGR
  - VBAT_STATUS – Battery voltage status (weak, good, or near end-of-charge)
  - IBAT_MAX – Maximum battery current setting
  - IBAT_SAFE – Maximum battery current setting (configurable once)
  - Charging status interrupts such as CHG_DONE, CHG_FAILED, etc.
  - VDD_MAX – Maximum battery voltage; typically 4.2 V or 4.35 V
  - VDD_SAFE – Maximum battery voltage (configurable once); typically 4.5 V
  - VIN_MIN – Minimum voltage to which the charger can be collapsed (must be 100 mV higher than VDD_MAX)
  - CHG_CTRL – Enable, disable or pause charging
  - VBAT_TRKL – ATC A threshold
  - VBAT_WEAK – ATC B software trickle charging threshold
  - IBAT_ATC_A – Maximum current during ATC A stage
  - IBAT_ATC_B – Maximum current during ATC B software trickle charging
  - VBAT_DET – CC to CV charging threshold

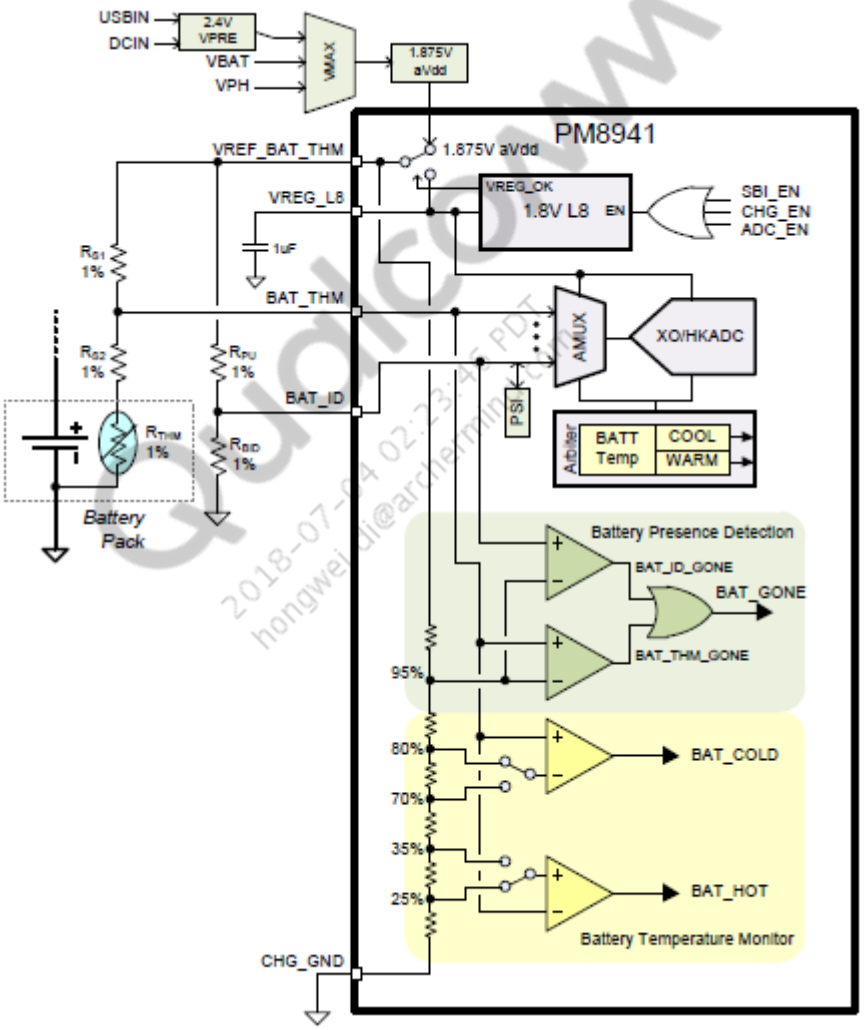
SMBB Charger Peripherals (cont.)

- SMBB_BUCK
  - Charger current voltage and battery current voltage monitoring loop status
  - Charging buck enable and disable settings
  - Charger current voltage and battery current voltage gain control settings
  - Charger buck current limit settings
  - Charger buck slew rate control

SMBB Charger Peripherals (cont.)

- SMBB_BAT_IF
  - Battery presence and battery thermistor/ID presence status
  - Battery temperature status
  - VREF_BAT_THM configuration
  - Battery FET status
  - Battery presence detection select between battery ID and thermistor
  - Battery temperature threshold adjustment
  - Internal BATFET regulator control

SMBB Charger Peripherals (cont.)



SMBB Charger Peripherals (cont.)

- SMBB_USB_CHGPTH
  - Power path selection and status
  - USB charger voltage validity
  - Charger presence status
  - AICL (Automatic Input Current Limiting) FSM status
  - USB charger over-voltage and under-voltage setting
  - Maximum USB charging current selection and enable
  - USB suspend setting if the USB connected is invalid or not enumerated
  - USB OTG (on-the-go) enable
  - USB enumeration timer setting and stop bit
  - Overcurrent protection settings
  - AICL debounce and delay settings

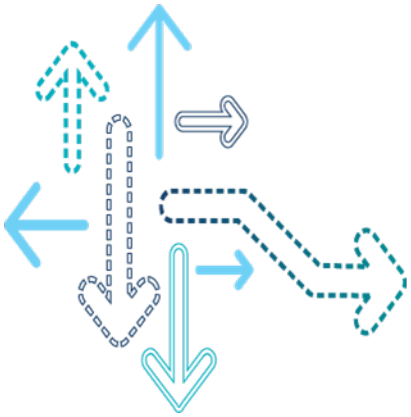
SMBB Charger Peripherals (cont.)

- SMBB_BOOST
  - Boost voltage selection for flash LED
  - Adaptive Boost mode to automatically adjust the voltage headroom
  - Boost Pass mode to bypass boost regulator
  - Boost regulator forced or hardware-controlled enable
  - Boost soft start enable
  - Boost voltage upper and lower limit settings
  - Boost voltage over-voltage protection control
  - Boost regulator parameters such as gain, max duty cycle, and so on

SMBB Charger Peripherals (cont.)

- SMBB_MISC
  - Adaptive boost control settings
  - Thermal Fault Tolerance (TFT) settings
- COIN
  - Coin cell resistance setting
  - Coin cell voltage setting
  - Coin cell charger enable/disable

QPNP Charger System

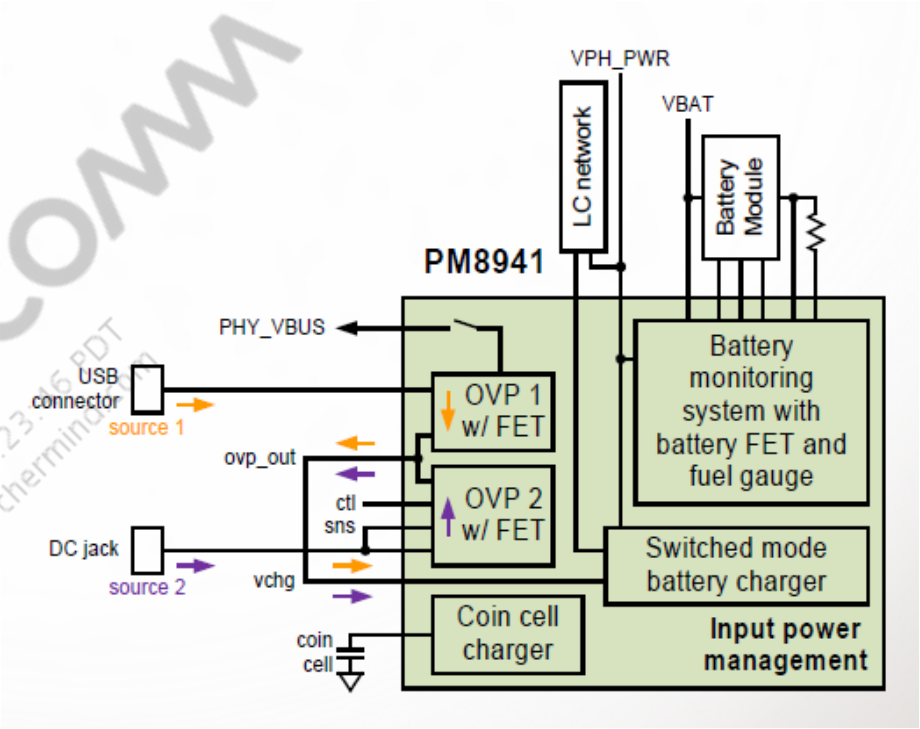


QPNP Charger

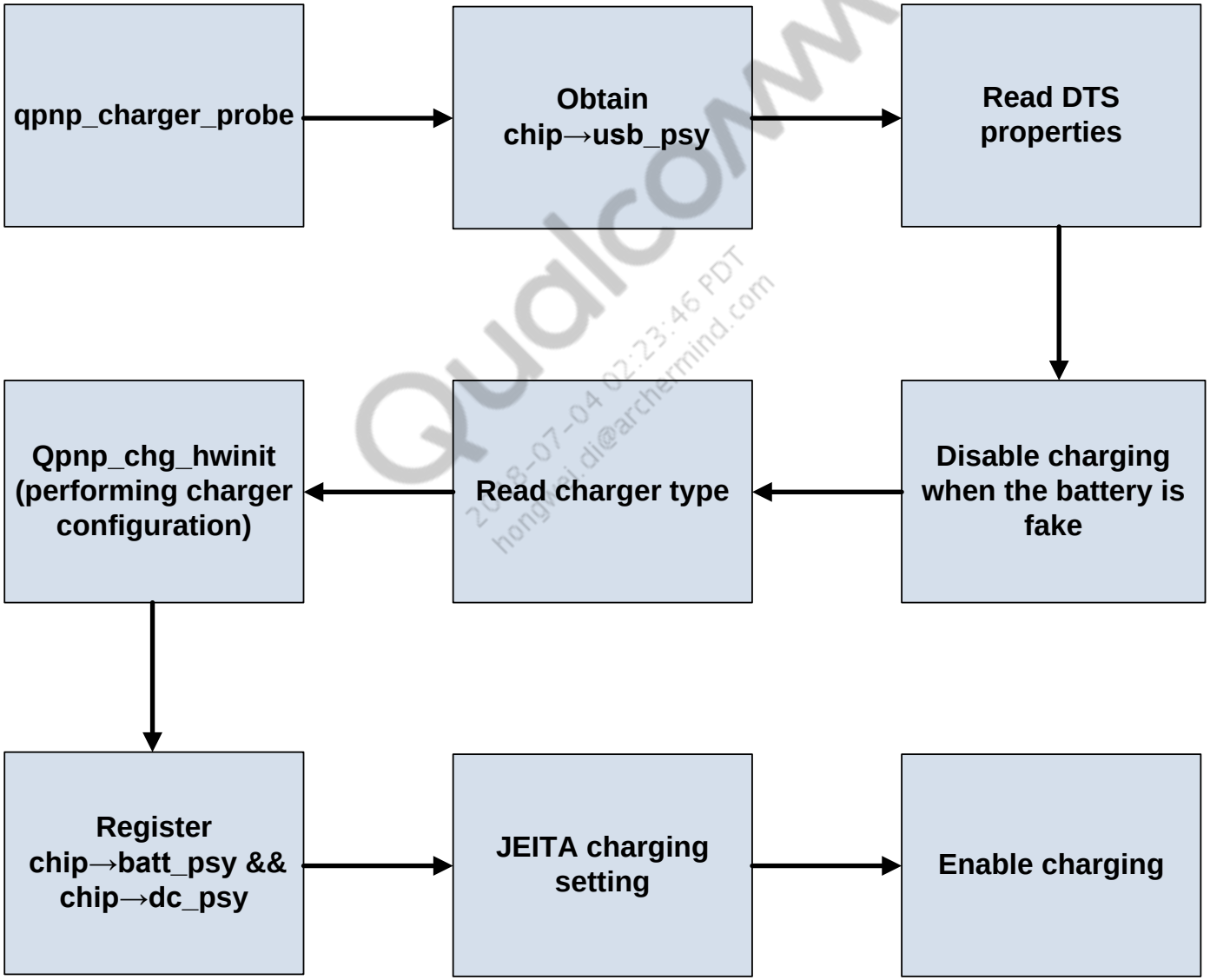
- When qpnp-charger is initiated, the probe() function is called first
- Then, qpnp_charger_probe() starts the parameter configuration indirectly through qpnp_chg_hwinit(), and then enables charging
- Several interrupt handlers are used to update the charger and battery status

Charger System

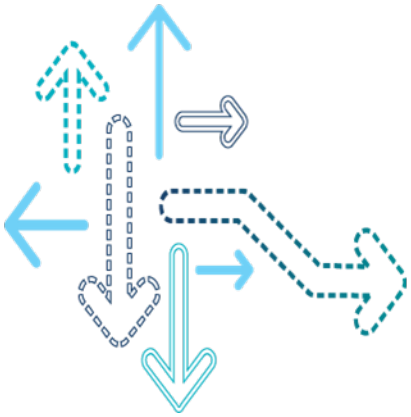
- USB charging path
  - Fully integrated 30 V OVP FET and control OVP register threshold [9.5 V - 11 V]
- DC charging path
  - Integrated 15 V OVP FET
- External charger detection
  - If the external charger is used, connect DC_IN_OVP_CTRL to GND
- Integrated BAT FET
  - Battery current senses across BAT_FET to eliminate the external R_Sense



QPNP Charger Initialization Flowchart



QPNP Charger Device Tree



QPNP Charger Device Tree

- The charger supports SMBB peripherals on QTI PMICs
- Each of these peripherals is implemented as subnodes in the msm-pm8941.dtsi file.

Peripheral	Description
qcom,chg-chgr	Supports charging control and status reporting
qcom,chg-bat-if	Battery status reporting, that is presence, temperature reporting, and voltage collapse protection
qcom,chg-buck	Charger buck configuration and status reporting with regard to several regulation loops, that is VDD, IBAT, and so on
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, that is buck frequency settings, comparator override features, and so on

QPNP Charger Device Tree (cont.)

- Parent node required properties

Property	Description
qcom,chg-vddmax-mv	Target battery voltage in mV
qcom,chg-vddsafe-mv	Maximum VDD voltage in mV
qcom,chg-vinmin-mv	Minimum input voltage in mV
qcom,chg-vbatdet-mv.	Battery charging resume voltage in mV
qcom,chg-ibatmax-ma	Maximum battery charge current in mA
qcom,chg-ibatterm-ma	Current at which the charging is terminated
qcom,chg-ibatsafe-ma	Safety battery current setting
qcom,chg-thermal-mitigation	Array of ibatmax values for different system thermal mitigation levels

QPNP Charger Device Tree (cont.)

- Parent node optional properties

Property	Description
qcom,chg-maxinput-usb-ma	Maximum input current; USB
qcom,chg-maxinput-dc-ma	Maximum input current; DC
qcom,chg-charging-disabled	Set this property to disable charging by default. The property can be overridden by the charging_enabled module parameter.
qcom,chg-use-default-batt-values	Set this flag to force-report the battery temperature of 250 decidegree Celsius, set the state of charge to 50%, and then disable charging

QPNP Charger Device Tree (cont.)

Subnode required properties

Property	Description
Compatible	Must be qcom,charger
reg	Specifies the SPMI address and size for the peripheral
Interrupts	Specifies the interrupt associated with the peripheral
interrupt-names	Specifies the interrupt names for the peripheral; every available interrupt must have an associated name with it to identify its purpose

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

                interrupt-names =       "chg-done",
                                        "chg-failed",
                                        "fast-chg-on",
                                        "trkl-chg-on",
                                        "state-change",
                                        "chgwdog",
                                        "vbat-det-hi",
                                        "vbat-det-lo";
};
```

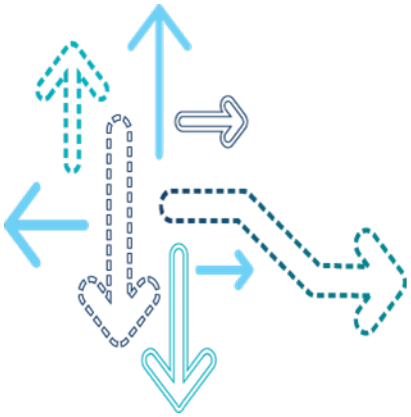
Snapshot of the Charger Device Tree Node

```
pm8941_chg: qcom,charger {
                        spmi-dev-container;
                        compatible = "qcom,qpnp-charger";
                        #address-cells = <1>;
                        #size-cells = <1>;
                        status = "disabled";

                        qcom,chg-vddmax-mv = <4200>;
                        qcom,chg-vddsafe-mv = <4200>;
                        qcom,chg-vinmin-mv = <4200>;
                        qcom,chg-ibatmax-ma = <1500>;
                        qcom,chg-ibatsafe-ma = <1500>;
                        qcom,chg-thermal-mitigation = <1500 700 600 325>;
                        qcom,chg-cool-bat-decidegc = <100>;
                        qcom,chg-cool-bat-mv = <4100>;
                        qcom,chg-ibatmax-warm-ma = <350>;
                        qcom,chg-warm-bat-decidegc = <450>;
                        qcom,chg-warm-bat-mv = <4100>;
                        qcom,chg-ibatmax-cool-ma = <350>;
                        qcom,chg-vbatdet-delta-mv = <350>;

…
}
```

Disable Charger

- If the qcom,chg-charging-disabled property is set in the node, it reads this property and disables the charger when the driver is initiated
- ADB steps to persistently disable the charging and limit USB current:
  1. Boot the phone
  2. adb root && adb wait-for-devices
  3. adb shell setprop persist.usb.chgdisabled 1
  4. adb sync
  5. Reboot the phone
- ADB steps to disable the charging and limit USB current:
  1. Boot the phone
  2. adb root && adb wait-for-devices
  3. adb shell "echo 1 > /sys/class/power_supply/battery/charging_enabled"
  4. adb shell sync

QPNP Charger Driver



Interface Between QPNP Charger and Device Tree

- The DTS property can be obtained by of_property_read_u32()

```
/**
  * of_property_read_u32- Find from a property
  * @np:         device node from which the property value is to be
  read.
  * @propname:   name of the property to be searched.
  * @out_value:       pointer to return value
  *
  * Returns 0 on success,
  * EINVAL if the property does not exist,
  * ENODATA if property does not have a value, and
  * EOVERFLOW if the property data isn't large enough.
*/

static inline int of_property_read_u32(const struct device_node *np,
                                       const char *propname,
                                       u32 *out_value)
```

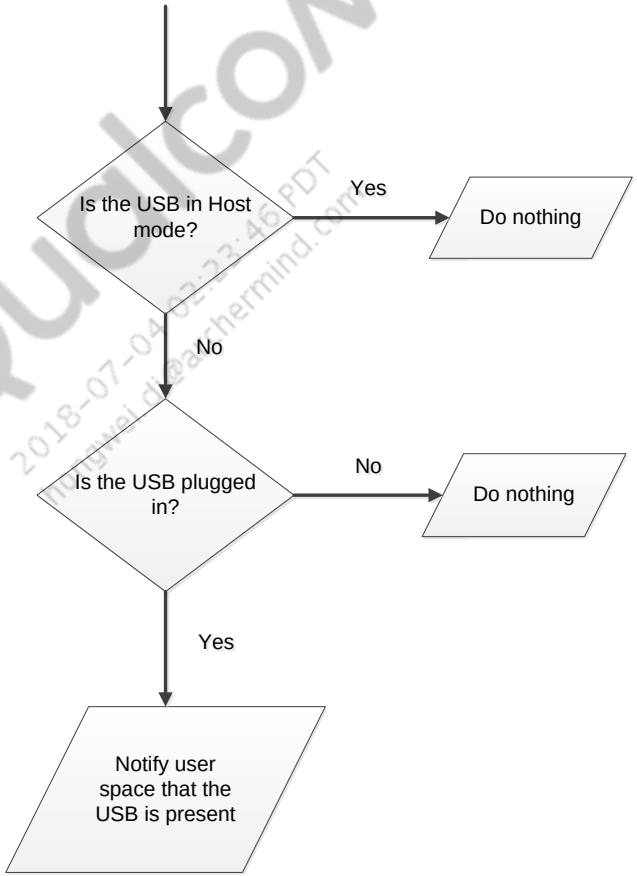
Interface Between QPNP Charger and Device Tree (cont.)

- struct qpnp_chg_chip
  - Stores the device information obtained from the device tree
- SPMI register access APIs
  - qpnp_chg_read(struct qpnp_chg_chip *chip, u8 *val, u16 base, int count)
  - qpnp_chg_write(struct qpnp_chg_chip *chip, u8 *val, u16 base, int count)
  - qpnp_chg_masked_write(struct qpnp_chg_chip *chip, u16 base, u8 mask, u8 val, int count)

QPNP Charger Interrupt Handler

qpnp_chg_usb_usbin_valid_irq_handler

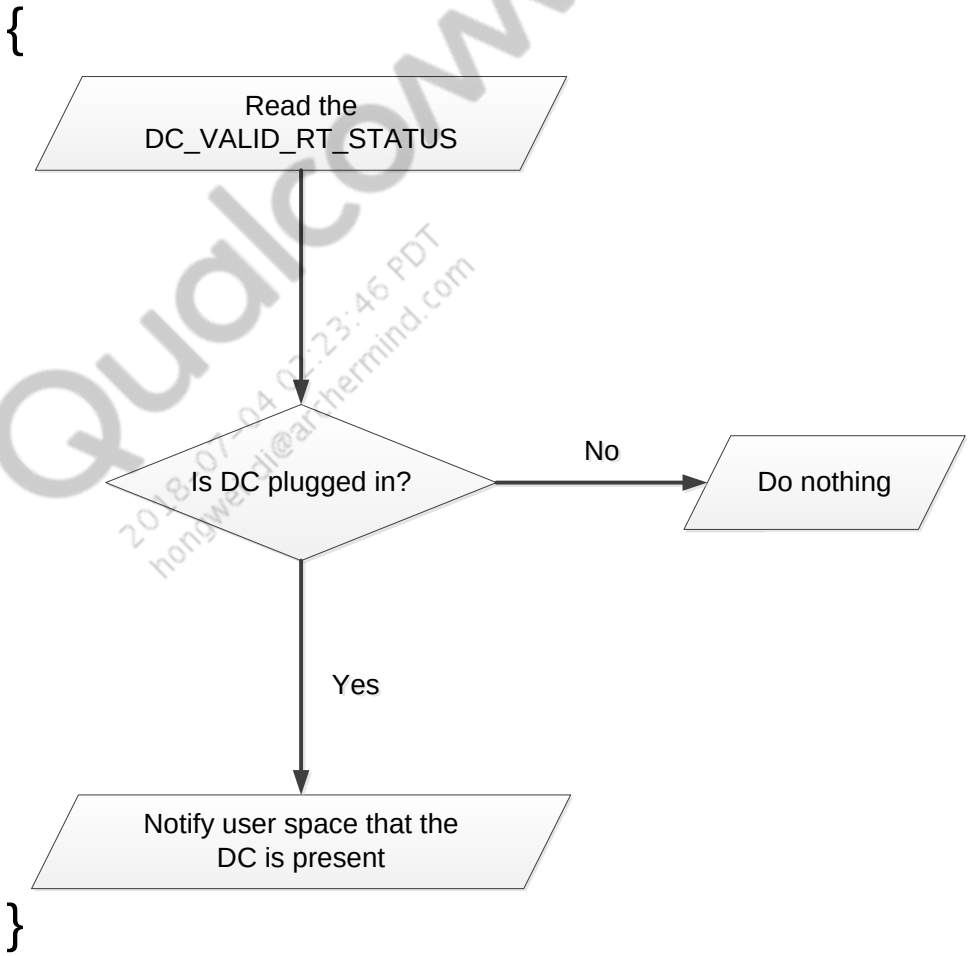
{



}

QPNP Charger Interrupt Handler (cont.)

- qpnp_chg_dc_dcin_valid_irq_handler



QPNP Charger Interrupt Handler (cont.)

- qpnp_chg_chgr_chg_failed_irq_handler(int irq, void *_chip)

```
{
   Clear CHGR_CHG_FAILED_BIT to make the charger work again.
}
```

  - Autofast charging fails due to expiration of the safety timer
  - Clear the failed bit to resume charging

- qpnp_chg_chgr_chg_done_irq_handler(int irq, void *_chip)

```
{
   Set chip->chg_done = true;
}
```

  - Autocharging has completed successfully
  - Set the chg_done flag to True

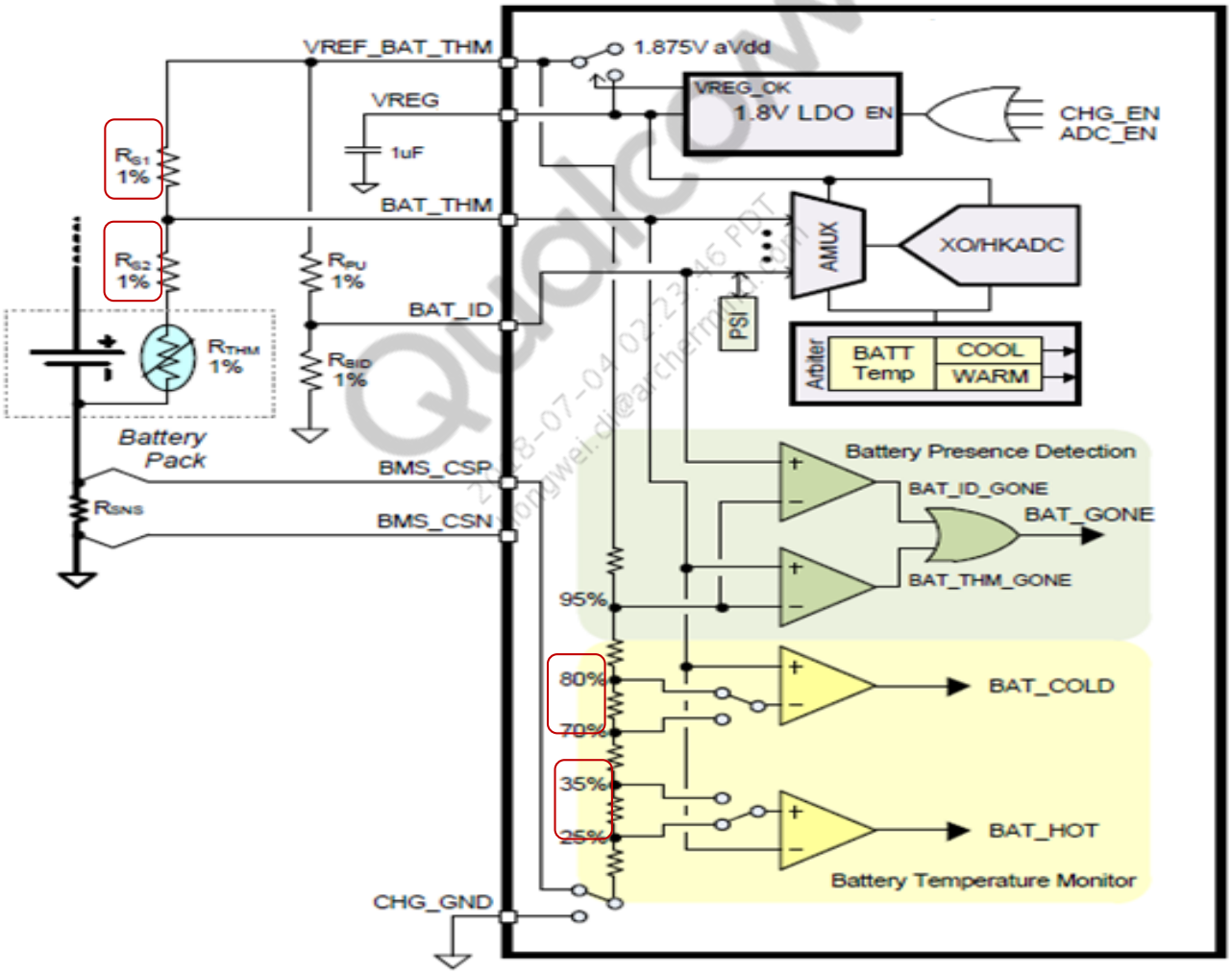
Other Interrupt Handlers

- qpnp_chg_bat_if_batt_pres_irq_handler()
  - Battery presence indicator
  - Update the battery status if the battery present status changes
- qpnp_chg_chgr_chg_failed_irq_handler()
  - Autofast charging fails due to the expiration of the safety timer
  - Clear the failed bit to resume charging
- qpnp_chg_chgr_chg_trklchg_irq_handler()
  - The interrupt is high when the linear trickle charger is on during software controlled charging
  - Set the chg_done flag to False
- qpnp_chg_chgr_chg_fastchg_irq_handler()
  - The charger is fast charging
  - Set the chg_done flag to False

USB Mode Control APIs

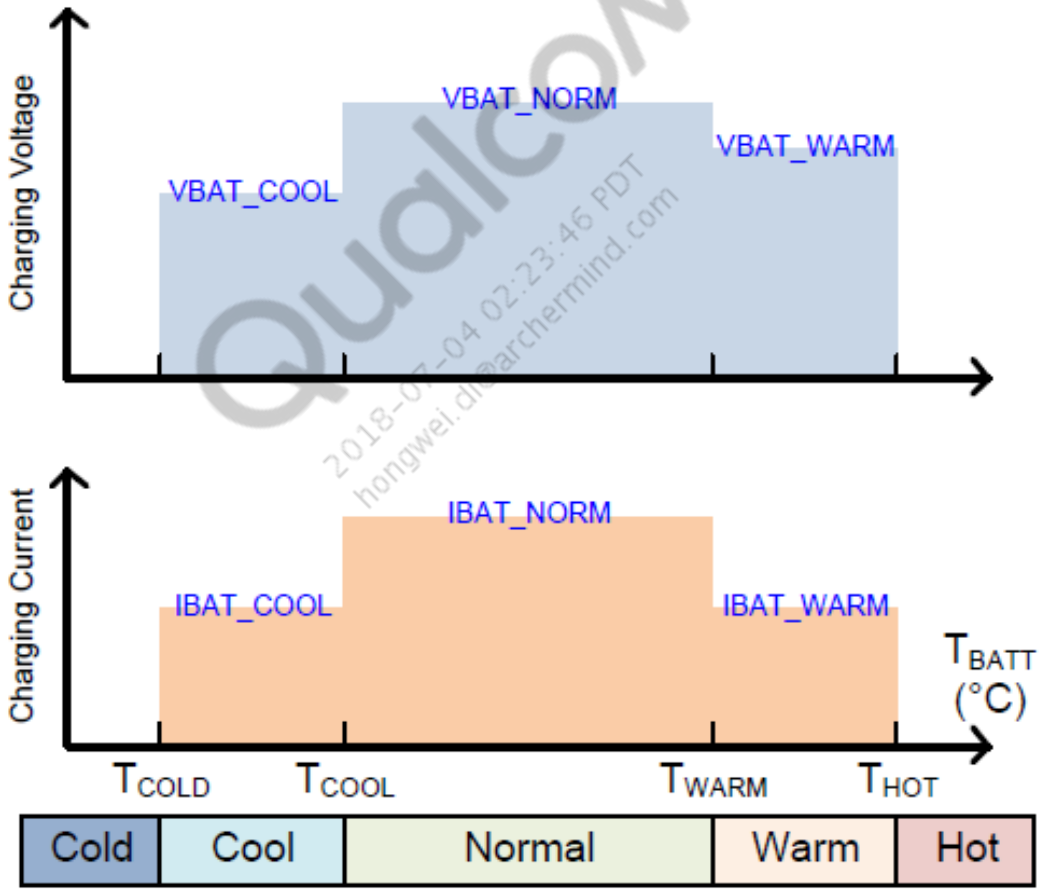
- USB Charge mode
  - switch_usb_to_charge_mode() – The system current is drawn from the charger
- USB Host mode
  - switch_usb_to_host_mode() – The system current is drawn from the battery
- USB suspension
  - qpnp_chg_usb_suspend_enable()

JEITA Charging (Not Currently Implemented)



JEITA Charging (Not Currently Implemented) (cont.)

- Temperature thresholds (JEITA compliance)



JEITA Charging APIs

- qpnp_chg_set_appropriate_vddmax()
  - If the battery is cool or warm, set vddmax to the predefined cool_bat_mv and warm_bat_mv respectively
- qpnp_chg_set_appropriate_vbatdet()
  - When the battery is cool, set vbatdet = cool_bat_mv – resume_delta_mv
  - If the battery is warm, set vbatdet = warm_bat_mv – resume_delta_mv
- The system can change the charging current based on its thermal level
  - Device tree property – qcom,chg-thermal-mitigation
    - Array of ibatmax values for different system thermal mitigation levels
  - qpnp_chg_set_appropriate_battery_current()
    - Set the ibatmax as the minimum current among cool_bat_chg_ma, warm_bat_chg_ma, and themal_mitigation[therm_lvl_del]

Select the Thermistor Pull-up Resistors (Rs1 and Rs2)

1. Find the battery thermistor parameters, room temperature, resistance (R0), and temperature coefficient (B)
2. Determine the allowable battery charging temperature range, and so on 0°C (TCOLD) to 40°C (THOT)
3. Calculate the thermistor resistance at cold and hot:
   - RCOLD = R0·exp(B·(1/TCOLD-1/To))
   - RHOT = R0·exp(B·(1/THOT-1/To))
4. Select the BTM comparator thresholds:
   - For a traditional battery charging temperature window, that is 0°C to 40/45°C, set the cold and hot thresholds to 70% and 35% respectively
   - For a JEITA-compliant extended battery charging temperature window, that is -10°C to 60°C, select 80% and 25% as the cold and hot thresholds respectively

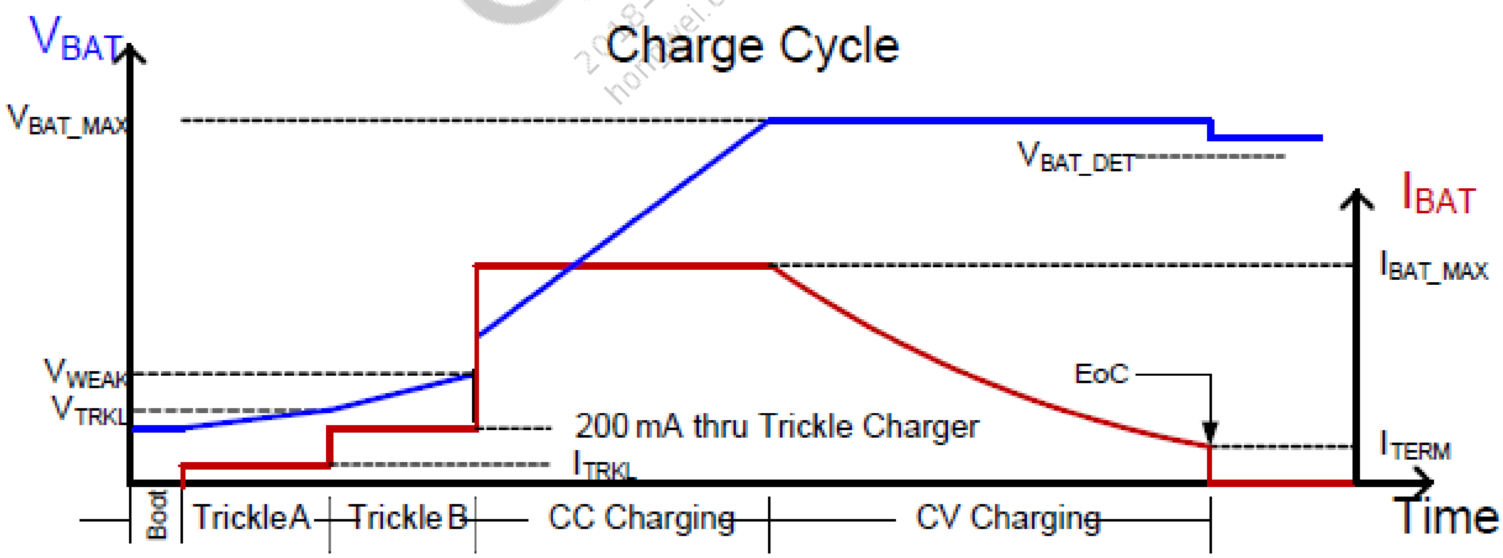
Select the Thermistor Pull-up Resistors (Rs1 and Rs2) (cont.)

Another possible root cause for why the OEM cannot start charging is, if the OEM selects the wrong resistor, the charger may not start because the PMIC incorrectly regards the temperature as too cold or too hot.

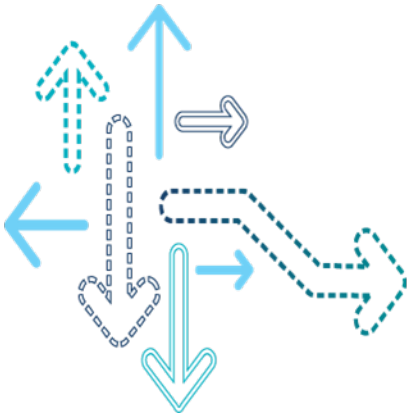
Better charging temperature window	BTM comparator thresholds	R_{s1} and R_{s2} calculation
0° to 40/45°C	70%/35%	$R_{s1} = 39(R_{COLD} - R_{HOT}) / 70$ $R_{s2} = (3R_{COLD} – 13R_{HOT})/ 10$
-10° C to 60°C	80%/25%	$R_{s1} = 3(R_{COLD} – R_{HOT}) / 11$ $R_{s2} = (R_{COLD} – 12R_{HOT})/ 11$

Charging Cycle

- The PMIC adopts hardware-controlled charging
- The driver configuration parameters in the kernel for fast charging are:
  - Constant current charging
  - Constant voltage charging
- When VBAT reaches the VBAT_DET threshold, the charging state changes from constant current charging to constant voltage charging
- Charging stops when IBAT is less than the ITERM threshold



Power Supply



Power Supply Framework to Export Information to User Space

- qpnp-charger has four power supply interfaces:
- Maintained by qpnp-charger.c
  - struct power_supply dc_psy; – Used to update the DC status
  - struct power_supply batt_psy; – Used to update the battery status
- Updated only by qpnp-charger.c
  - struct power_supply usb_psy; – Used to update the USB status
  - struct power_supply bms_psy; – Indirectly calls the APIs implemented in qpnp-bms.c

QPNP Charger – Update Power Supply Interfaces

- power_supply_changed(&chip->dc_psy);
  - qpnp_chg_dc_dcin_valid_irq_handler()
- power_supply_set_present(chip->usb_psy, chip->usb_present);
  - qpnp_chg_usb_usbin_valid_irq_handler()
- power_supply_set_present(chip->bms_psy, batt_present)
  - qpnp_chg_bat_if_batt_pres_irq_handler
- power_supply_changed(&chip->batt_psy);
  - qpnp_chg_bat_if_batt_pres_irq_handler()
  - qpnp_chg_chgr_chg_trklchg_irq_handler()
  - qpnp_chg_chgr_chg_fastchg_irq_handler()
  - qpnp_chg_chgr_chg_done_irq_handler()

Power Supply Parameters

- dc_psy

```
static enum power_supply_property pm_power_props_mains[] = {
  POWER_SUPPLY_PROP_PRESENT, //<- Indicates charger present
  POWER_SUPPLY_PROP_ONLINE, //Indicates charger is online
  POWER_SUPPLY_PROP_CURRENT_MAX, // Indicates programmed maximum charger
  current
};
```

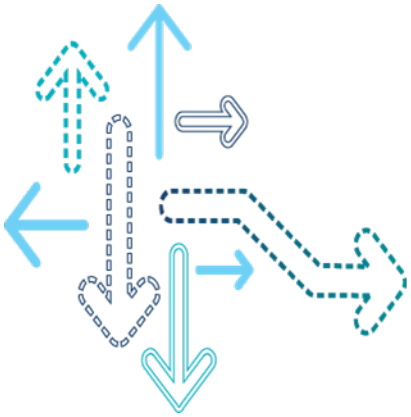
- Difference between online and present property
  - Online represents whether the charger is active or not
  - Present represents whether the charger is connected or not
- For example, whenever a USB charger is connected, it becomes present and whenever this port is successfully enumerated, it becomes online

Power Supply Parameters (cont.)

- batt_psy

```
static enum power_supply_property msm_batt_power_props[] = {
   POWER_SUPPLY_PROP_CHARGING_ENABLED, //Indicates whether charging is enabled
   POWER_SUPPLY_PROP_STATUS,           //Indicates whether battery is charging/discharging
   POWER_SUPPLY_PROP_CHARGE_TYPE,   //Indicates whether fast or trickle charging
   POWER_SUPPLY_PROP_HEALTH,           //Indicates whether battery is warm/cold/normal
   POWER_SUPPLY_PROP_PRESENT,          //Indicates whether battery is connected
   POWER_SUPPLY_PROP_ONLINE,           //Indicates whether battery is open or connected
   POWER_SUPPLY_PROP_TECHNOLOGY,     //Indicates LiON type battery
   POWER_SUPPLY_PROP_VOLTAGE_MAX_DESIGN, //Indicates maximum battery voltage
   POWER_SUPPLY_PROP_VOLTAGE_MIN_DESIGN,  //Indicates minimum usable voltage
   POWER_SUPPLY_PROP_VOLTAGE_NOW,              //Indicates battery voltage
   POWER_SUPPLY_PROP_CAPACITY,                    //Indicates battery SOC capacity
   POWER_SUPPLY_PROP_CURRENT_NOW,          //Indicates current flowing through battery
   POWER_SUPPLY_PROP_INPUT_CURRENT_MAX, //Indicates maximum current limit on battery
   POWER_SUPPLY_PROP_VOLTAGE_MIN,             //Indicates minimum charger voltage
   POWER_SUPPLY_PROP_INPUT_VOLTAGE_REGULATION, //Indicates status of VCHG loop
   POWER_SUPPLY_PROP_CHARGE_FULL_DESIGN,          //Indicates max capacity in mAh
   POWER_SUPPLY_PROP_CHARGE_FULL,                    //Indicates whether charge is full
   POWER_SUPPLY_PROP_TEMP,                              //Indicates battery temperature
   POWER_SUPPLY_PROP_COOL_TEMP,                     //Indicates cool temperature setting
   POWER_SUPPLY_PROP_WARM_TEMP,                    //Indicates warm temperature setting
   POWER_SUPPLY_PROP_SYSTEM_TEMP_LEVEL,        //Indicates current thermal level of battery
   POWER_SUPPLY_PROP_CYCLE_COUNT,                  //Indicates number of charging cycles
};
```


QPNP Charging Algorithm



QPNP Charging Algorithm

- Charging algorithm is based on the following important works:
  - adc_measure_work (qpnp_bat_if_adc_measure_work())
  - reduce_power_stage_work (qpnp_chg_reduce_power_stage_work())
  - adc_disable_work (qpnp_bat_if_adc_disable_work())
  - eoc_work (qpnp_eoc_work())
  - arb_stop_work (qpnp_arb_stop_work())
  - soc_check_work (qpnp_chg_soc_check_work())
  - aicl_check_work (qpnp_aicl_check_work())

QPNP Charging Algorithm – adc_measure_work

- adc_measure_work
  - Usage – The adc_measure_work work is used to configure the battery temperature of the ADC channel. An interrupt is triggered if the battery temperature goes beyond thresholds and qpnp_chg_adc_notification() is called. ADC continuously measures the battery thermistor channel in the hardware to see if the temperature is going beyond thresholds.
  - Parameters to configure
    - DTSI
      - `qcom,warm-bat-decidegc = <600>; //warm battery temperature 60degc`
      - `qcom,cool-bat-decidegc = <60>; //cool battery temperature 6degc`
    - qpnp-charger.c
      - `chip→adc_param.timer_interval = ADC_MEAS2_INTERVAL_1S; // measurement interval configured between 0 to 1500ms in 100ms steps`
      - `chip→adc_param.threshold_notification = qpnp_chg_adc_notification; //Callback function if ADC readings crosses warm/cool threshold`
      - `chip→adc_param.channel = LR_MUX1_BATT_THERM; //Battery thermistor channel`
  - Result
    - qpnp_chg_adc_notification() checks if the interrupt is triggered because of cool or warm condition
    - Switches off the charging if the temperature is outside cool or warm threshold
    - Configures VDDMAX, IBAT_MAX, and VBATDET according to the battery temperature state
    - If the battery temperature becomes warm/cool, then these parameters are reduced to avoid further increase in the battery temperature

QPNP Charging Algorithm – reduce_power_stage_work

- reduce_power_stage_work
  - Usage – The reduce_power_stage_work work runs every 20 sec to check the charging buck regulator's power stage. It optimizes the power stage for the buck regulator when all of the following conditions are True:
    - USB charging current configured is over the maximum limited current (USB_WALL_THRESHOLD_MA)
    - Battery voltage is below POWER_STAGE_REDUCE_MAX_VBAT_UV
    - Charger voltage is above POWER_STAGE_REDUCE_MIN_VCHG_UV or charger voltage drops below VIN_MIN threshold
  - Parameters to configure
    - qpnp-charger.c
      - `USB_WALL_THRESHOLD_MA //Maximum current expected from USB charger`
      - `POWER_STAGE_REDUCE_MAX_VBAT_UV // Maximum battery voltage up to which power stage reduction should be done`
      - `POWER_STAGE_REDUCE_MIN_VCHG_UV //Minimum charger voltage upto which power stage reduction should be done`
      - `POWER_STAGE_REDUCE_CHECK_PERIOD_SECONDS //Period of repeating reduce_power_stage_work`

QPNP Charging Algorithm – reduce_power_stage_work (cont.)

- Result
  - qpnp_chg_reduce_power_stage() checks if power stage needs to be reduced or restored
  - If power stage needs to be reduced or restored, qpnp_chg_power_stage_set() is called
  - This ensures that the buck operates in reduced or restored stage only when it is required
  - The buck provides minimum required power in reduced stage to avoid extra power consumption and generate less thermal heat

QPNP Charging Algorithm – adc_disable_work

- adc_disable_work
  - Usage – The adc_disable_work work runs whenever the battery is removed to switch off the battery temperature of the ADC channel. This work is a complement of adc_measure_work and is triggered from qpnp_chg_bat_if_batt_pres_irq_handler(), which indicates whether a battery is removed or connected.
  - Result
    - qpnp_bat_if_adc_disable_work() calls qpnp_adc_tm_disable_chan_meas(chip→adc_tm_dev, and chip→adc_param) to switch off the battery temperature of the ADC channel
    - Reduces current consumed in measuring the ADC values for battery temperature as the readings are stopped when the battery is not connected

QPNP Charging Algorithm – eoc_work

- eoc_work
  - Usage – The eoc_work work runs every 10 sec to check if the charging is coming to an end
    - Increases or decreases VDD_MAX during CV stage of charging to extract maximum charging current and charge quickly
    - Decides when the charging must resume after charging has finished
    - eoc_work is scheduled whenever a DC or USB charger is connected
    - Starts when VBATDET_LO or FASTCHG IRQ is triggered to ensure that this work is running
    - eoc_work also holds the wakelock until it finishes execution
  - Parameters to configure
    - DTSI
      - `qcom,vddmax-mv = <4200>; // Maximum voltage up to which the battery is charged`
      - `qcom,vbatdet-delta-mv = <100>; //Voltage less than VDDMAX at which charging should resume`
      - `qcom,ibatterm-ma= <100>; //Minimum current at which battery can be considered fully charged`
      - `qpnp-charger.c`
      - `VBATDET_MAX_ERR_MV //Maximum error margin in VBATDET setting`
      - `CONSECUTIVE_COUNT //Number of times IBAT readings needs to below ibatterm to decide end of charging`
      - `MAX_DELTA_VDD_MAX_MV // Maximum voltage by which VDD_MAX to be increased`

QPNP Charging Algorithm – eoc_work (cont.)

- Result
  - qpnp_eoc_work() decides that charging has finished when all the following three conditions are fulfilled:
    - Battery voltage is not above VDD_MAX (that is, the device is in CV charging)
    - IBAT is less than ibatterm more than three times
    - Discharging did not start between three consecutive iterations
  - Once the end of charging is decided, the battery power supply is changed to indicate that the battery is full. The charging is disabled and enabled again to stop charging until VBAT reaches below VBATDET.
  - qpnp_eoc_work() continues running to check when the charging should resume
  - Charging resumes when VBAT is less than chip→max_voltage_mv - chip→resume_delta_mv - VBATDET_MAX_ERR_MV for more than three times
  - Once the above condition is met, VBATDET_LOW IRQ is enabled again so that the IRQ handler can update battery power supply status to charging
  - qpnp_eoc_work() also tries to maximize CV stage charging current to a higher value by increasing VDD_MAX using qpnp_chg_adjust_vddmax()
  - Maximum limit to increase VDD_MAX is MAX_DELTA_VDD_MAX_MV

QPNP Charging Algorithm – arb_stop_work

- arb_stop_work
  - Usage – The arb_stop_work work runs whenever CHG_GONE IRQ is triggered. CHG_GONE IRQ indicates that the charger is disconnected and goes to qpnp_chg_usb_chg_gone_irq_handler() to avoid reverse boost from the battery to the charger. The IRQ handler then switches off charging and runs the system on battery for 1 sec. The charging is then enabled again in qpnp_arb_stop_work().
  - Parameters to configure
    - qpnp-charger.c
      - `#define ARB_STOP_WORK_MS 1000 //Period after which reverse boost worker should run`
  - Result
    - This work is used to avoid reverse boost from battery to charger whenever the charger is removed
    - It disables charging and waits for 1 sec to enable charging again
    - It also refreshes the charger FSM and ensures that the charger does not go into unintentional reverse boost

QPNP Charging Algorithm – soc_check_work

- soc_check_work
  - Usage – The soc_check_work work runs whenever a USB or DC charger is connected to the phone. It calls get_prop_capacity() to update the battery and charger power supplies to update the UI. The soc_check_work work also restarts charging if the battery SOC is below 95%, which is programmable as follows:
  - Parameters to configure
    - DTSI
      - `qcom,resume-soc = <95>; //SOC at which charging should resume`
    - qpnp-charger.c
      - `#define DEFAULT_CAPACITY 50 //Default capacity to show if battery is not connected`

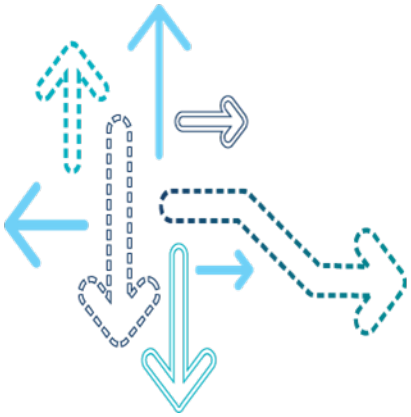
QPNP Charging Algorithm – soc_check_work (cont.)

- Result
  - soc_check_work ensures that the charging restarts if all the following conditions are met:
    - Charger is inserted
    - Charging has been disabled before
    - SOC is below 95%
  - soc_check_work then sets VBATDET to (chip→max_voltage_mv+chip→resume_delta_mv) and enables charging
  - The difference between resume charging with respect to eoc_work is as follows:
    - eoc_work resumes charging if the charger is still connected after the charging is finished and battery voltage drops below VBATDET
    - soc_check_work resumes charging if the charger is newly connected after the charging is finished

QPNP Charging Algorithm – aicl_check_work

- aicl_check_work
  - Usage – The aicl_check_work work is invoked from qpnp_charger_probe() during phone bootup. It is used to check if AICL is running from user space. If AICL is not present, aicl_check_work directly sets the IUSB_MAX current to the PMIC register to draw maximum current from the USB.
  - Result – aicl_check_work checks whether a device has AICL algorithm or not

QPNP Charger Additional Features



QPNP Charger Additional Features – IR Drop Compensation

- During fast charging, there is a significant drop across VPH_PWR and VBAT due to resistance from battery FET RDS ON and trace routing. This reduces the end-of-charging capacity by a significant amount. For example, for a 3A current with 100 MΩ of resistance, the drop is 300 mV which might end the charging prematurely.
- There are two ways to mitigate this problem
  - Analog IR drop compensation (default) – This mode of IR drop compensation automatically regulates the VBAT pin instead of VPH_PWR and adjusts VDD_MAX to end the charging properly
  - Digital IR drop compensation – In digital IR drop compensation, BMS reads the battery current and this current can then be multiplied by the resistance factor measured during calibration to adjust VDD_MAX

Note: Compensation takes a finite battery resistance value into account while analog IR compensation automatically adjusts VDD_MAX depending on the actual battery resistance, which is subject to change by temperature.

QPNP Charger Additional Features – IR Drop Compensation (cont.)

- To enable digital IR drop compensation, the analog IR drop compensation should be disabled in

```
rc = qpnp_chg_masked_write(chip, chip→buck_base + CHGR_BUCK_BCK_VBAT_REG_MODE,
BUCK_VBAT_REG_NODE_SEL_BIT, BUCK_VBAT_REG_NODE_SEL_BIT, 1);
```

to

```
rc = qpnp_chg_masked_write(chip,
                  chip->buck_base + CHGR_BUCK_BCK_VBAT_REG_MODE,
                  BUCK_VBAT_REG_NODE_SEL_BIT,
                  0, 1);
in qpnp_chg_hwinit()
```

  - Also, the battery resistance must be programmed into 0x1067 SMBBP_CHGR_IR_DROP_COMPEN register bit 3:0 in units of 36 MΩ

SMBBP_CHGR_I R_DROP_COMPEN

Bits	Name	Type	Description
7	EN	RW	IR drop compensation enable during fast charging 0 = disabled 1 = enables automatic adjustment of VDD_MAX using BMS IBAT data
3:0	RES	RW	Resistance of charger buck output to battery pack 36 mΩ * X

QPNP Charger Additional Features – Charger Maximum Temperature Configuration

- Charging can be stopped automatically when the charger reaches a particular temperature to avoid too much heat. The current configuration is 120°C to stop charging and 95°C to resume charging when the charger temperature returns to normal. These temperatures are configurable from 75–150°C in the 0x1066 register.

SMBBP_C HGR_CHG_TEM P_THRESH

Bits	Name	Type	Description
7:4	CHG_TEMP_STOP	RW	T = 75°C + (X * 5°C) Maximum charging temperature in hysteresis mode: charging stops and pass device turns off at this temperature
3:0	CHG_TEMP_ RESUME	RW	T =75 C + (X • 5 °C) In hysteresis mode, pass device turns on, charging resumes when the pass device has fallen below this temperature.

References

Acronyms	
Term	**Definition**
AICL	Automatic Input Current Limiting
BTM	Battery Threshold Monitoring Battery Temperature Monitoring
FSM	Finite State Machine
QPNP	Qualcomm Plug-n-Play
SMBB	Switch Mode Battery Charger and Boost
SOC	State Of Charge

Questions?

https://support.cdmatech.com