Qualcomm Technologies International, Ltd.

Audio Sink Application Power Management

User Guide

80-CT450-1 Rev. AD

October 19, 2017

Confidential and Proprietary – Qualcomm Technologies International, Ltd.

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 International, Ltd. 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 International, Ltd.

Qualcomm BlueCore is a product of Qualcomm Technologies International, Ltd. Other Qualcomm products referenced herein are products of Qualcomm Technologies International, Ltd.

Qualcomm is a trademark of Qualcomm Incorporated, registered in the United States and other countries. BlueCore is a trademark of Qualcomm Technologies International, Ltd., 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 International, Ltd. (formerly known as Cambridge Silicon Radio Limited) is a company registered in England and
Wales with a registered office at: Churchill House, Cambridge Business Park, Cowley Road, Cambridge, CB4 0WZ, United Kingdom.
Registered Number: 3665875 | VAT number: GB787433096

Revision history

Revision	Date	Description
1	JUN 2015	Original publication of this document. Alternative document number CS-00333471-UG.
2	SEP 2016	Updated to conform to QTI standards; No technical content was changed in this document revision
3	APR 2017	Updated for ADK 4.2. Added to the Content Management System.
AD	OCT 2017	Document Reference Number updated to use Agile number. No change to the technical content.

Contents

Tables

Figures

Equations

1 ADK power management - overview

The Power Management feature of the Audio Sink application is split over two layers, the Power library and the application. This document describes how the Power Management feature has been split and what responsibilities each layer takes in managing the power source.

The Audio Sink application can support a battery, for this, it manages the discharge and charging of the battery.

This document should be used with the *Audio Sink Application User Guide*.

2 Power library

The main function of the power library is to monitor voltage levels and notify the application if any predefined thresholds are crossed.

2.1 PowerInit function

The client task and configuration information are passed to the PowerInit function:

```
PowerInit(Task clientTask, const power_config *power_data)
```

The initial charger status and readings for each voltage to be monitored are sent to the client in a POWER_INIT_CFM message.

NOTE A voltage is considered under a threshold if the voltage is less than the configured threshold. When a voltage crosses a threshold a message is sent to the client to indicate that the voltage has crossed the threshold.

Voltages are read using the ADC and configured using the power_config parameter of PowerInit.

Table 2-1 Data structure of the power_config typedef

Type	Name	Size	Description
power_vref_config	vref	2 words	V_{REF} Configuration
power_vbat_config	vbat	8 words	V_{BAT} Configuration
power_vthm_config	vthm	10 words	V_{THM} Configuration
power_vchg_config	vchg	3 words	V_{CHG} Configuration

The components of the power_config structure are described in:

V_{REF} in the power_config parameter

V_{BAT} in the power_config parameter

V_{THM} in the power_config parameter

V_{CHG} in the power_config parameter

2.1.1 V_{REF} in the power_config parameter

As the voltage for the V_{REF} for a given IC is known, these readings are used to convert the readings from the ADC to a voltage reading in millivolts. The voltage is calculated using the equation shown in Equation 2-1 .

$$V_{\mathrm{Re}\,ading}(mV) = V_{\mathrm{Re}\,ading}(counts).\frac{V_{REF}(mV)}{V_{REF}(counts)}$$

Equation 2-1 ADC to millivolt conversion

The `power_adc` structure configures the ADC source to be read, the time between readings when the charger is attached and when the charger is detached. The `power_adc` structure is used in the configuration of all voltage readings.

Table 2-2 Data structure of the power_adc typedef

Type	Name	Size	Description
`vm_adc_source_type`	`source`	8 bit	ADC source to monitor
unsigned	`period_chg`	8 bit	Time between readings when charger attached (seconds)
unsigned	`period_no_chg`	1 word	Time between readings when charger detached (seconds)

2.1.2 V_{BAT} in the power_config parameter

VBAT readings are configured using `power_vbat_config`.

Table 2-3 Structure of the power_vbat_config typedef

Type	Name	Size	Description
unsigned	`notify_period`	8 bit	Interval between notifications to the client task (seconds) when voltage under this limit. If set to `0` a notification is only sent when the threshold is initially crossed.
unsigned	`limit`	8 bit	Voltage threshold (mV/20)

power_vbat_limit

`power_vbat_limit` allows up to six voltage notification thresholds to be configured. Thresholds are given in order of ascending voltage, when a threshold is crossed the client is sent a `POWER_BATTERY_VOLTAGE_IND` message with the index of the new threshold and the new voltage reading.

A threshold with a limit of 0xff should always be used to terminate the limits array. This entry is used to control the notification period for the sixth threshold index. Table 2-4 shows how each threshold is structured.

Table 2-4 Structure of each vbat threshold

Type	Name	Size	Description
unsigned	notify_period	8 bit	Interval between notifications to the client task (seconds) when voltage under this limit. If set to 0 a notification is only sent when the threshold is initially crossed.
unsigned	limit	8 bit	Voltage threshold (mV/20)

2.1.3 V_{THM} in the power_config parameter

power_vthm_limit allows up to seven voltage notification thresholds to be configured plus the other options described in Table 2-5.

Table 2-5 Structure of power_vthm_limit typedef

Type	Name	Size	Description
power_adc	adc	2 words	ADC Configuration, see Table 2-2
unsigned	delay	4 bit	Delay between setting PIO and reading V_{THM} (ms)
unsigned	unused	5 bit	Unused
unsigned	raw_limits	1 bit	Configure limits in ADC counts rather than mV
unsigned	drive_pio	1 bit	Drive thermistor by a PIO
unsigned	pio	5 bit	PIO to drive thermistor (0-31)
unsigned [7]	limits	7 words	Array of voltage thresholds in mV (or ADC counts if raw_limits is set). **NOTE** The list of limits should be terminated with 0xffff if less than seven limits are used.

Thermistor AIO

Thermistor readings are typically taken from an external AIO. On Qualcomm® BlueCore™ 5 and CSR6xxx external AIO readings saturate above 1.5 V, on BlueCore7 readings saturate above 1.35 V. It is important to ensure all thermistor limits fall below these values.

Thermistor PIO

In some cases, it may be necessary to drive the thermistor from a PIO. To do this the drive_pio flag must be set, and the pio field set to the correct PIO. To allow the thermistor reading to stabilize it may be necessary to delay reading the ADC for a short time after setting the PIO, this can be done using the delay field, up to a maximum delay of 15 ms.

Limits

As for battery voltage thresholds these are given in order of ascending voltage, when a threshold is crossed the client is sent a POWER_BATTERY_TEMPERATURE_IND message with the index of the new threshold and the new voltage reading. If the voltage reading is greater than all configured thresholds, then an indication with index eight is sent. By default limits are configured in mV and can be calculated from the thermistor's data sheet. Assuming a simple potential divider setup as shown in Figure 2-1 and Figure 2-3, where R is a fixed resistor and T is the resistance of the thermistor for a given temperature.

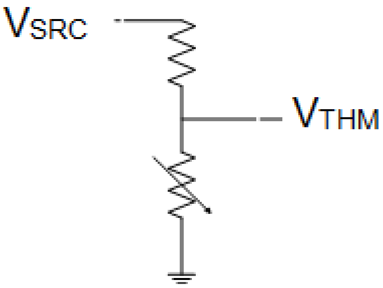

Figure 2-1 Simple potential divider setup

$$V_{THM}(mV) = \frac{T \times V_{SRC}(mV)}{T + R}$$

Equation 2-2 Calculation of V_{THM}

Raw limits

Where V_{SRC} is the same source as the ADC there is the option of configuring `limits` in ADC counts, this can be more accurate as it removes the need to calculate the voltage using V_{REF}

$$V_{THM}(counts) = \frac{T \times V_{SRC}(counts)}{T + R}$$

Equation 2-3 V_{THM} in ADC Counts

The Equation 2-3 can be used because it is known that V_{SRC} will be the maximum reading, that is 1023 counts.

2.1.4 V_{CHG} in the power_config parameter

`power_vchg_limit` allows configuration of a single voltage notification threshold, see Table 2-6.

Table 2-6 Structure of the power_vchg_limit typedef

Type	Name	Size	Description
`power_adc`	`adc`	2 words	ADC Configuration, see Table 2-2
unsigned	`limit`	1 word	voltage threshold in mV

If the threshold set in `limit` is crossed the client is sent a `POWER_CHARGER_VOLTAGE_IND` message indicating the new level, (`0` if below the threshold, `1` if above), and the new voltage reading.

2.2 Charger status

The power library polls the charger status every second until it detects that the charger has been removed. To resume monitoring of the charger status the client must call `PowerChargerMonitor` when it detects that the charger has been attached. The library notifies the client of any changes in charger status by sending a `POWER_CHARGER_STATUS_IND` message.

2.3 PowerChargerEnable

`PowerChargerEnable` allows the client to enable/disable the charger.

2.4 PowerChargerSetCurrent

`PowerChargerSetCurrent` allows the client to set the charge current. On BlueCore7 and later the value passed should be the current in mA to draw. Pre-BlueCore7 the value passed is a level (0-15) that is device dependent.

2.5 PowerChargerSetBoost

`PowerChargerSetBoost` allows the client to enable/disable boost charge. This updates the library boost enable flag to:

- `power_boost_disabled`
- `power_boost_internal_enabled`

or

- `power_boost_external_enabled`

When the charger monitor detects a change to Fast Charge state internal or external boost, charge is enabled based on the flag. If the client later disables boost charge the boost cycle is considered complete by the library.

Further calls to this function update the flag, but the charger monitor does not enable boost charge again until it detects the charger has been detached, at which point the boost cycle is reset.

2.6 PowerChargerSetVterm

This allows the client to set the charger termination voltage. This fails if the requested voltage is greater than the configured termination voltage. Passing a value of zero resets the charger termination voltage to the configured value. This should be used with BlueCore7 and later

3 Headset power manager

Power management provided for the ADK Headset application is described in:

- V_{BAT}
- V_{THM}
- Configuration of headset power settings
- Charger status

3.1 V_{BAT}

When the Headset receives a `POWER_BATTERY_VOLTAGE_IND` message, it applies the battery setting associated with the battery level. Table 3-1 describes how each battery level event is structured.

Table 3-1 headset_charge_setting

Type	Name	Size	Description
Unsigned	unused	5 bit	Reserved
`battery_level_source`	sources	3 bit	Bit mask that controls whether the event is generated based on the originator of the reading: ■ B001 - Generate the event for automatic readings ■ B010 - Generate the event for user initiated readings ■ B100 - Generate the event for initial reading
unsigned	`event`	8 bit	The event to generate for this battery level

Up to six events, (one for each power library threshold), can be configured; these should be selected from:

- `EventCriticalBattery` (`0xad`): Should be generated when the voltage drops to the point the headset needs to power off to preserve remaining power. This causes the Headset to power off unless the charger is connected.
- `EventLowBattery` (`0x1a`): Should be generated when the voltage drops to the point that low power notifications for the user should be generated. This generates a low battery warning tone/LED indication if configured to do so, unless the charger is connected.
- `EventGasGauge0` (`0x41`): Should be generated when the voltage drops to the point the Headset needs to activate intelligent power management. Sends CSR battery level = 2.
- `EventGasGauge1` (`0x42`): Sends CSR battery level = 4.

- `EventGasGauge2` (`0x43`): Sends CSR battery level = 7.
- `EventGasGauge3` (`0x44`): Sends CSR battery level = 9.

3.2 V_{THM}

When the Headset receives a `POWER_BATTERY_TEMPERATURE_IND` message, it adjusts the charger settings based on the new temperature level. Up to eight charger settings, (one for each possible index), can be configured. Each charger setting is made up of a charge current and charge termination setting. Table 3-2 shows how Charger Current settings are structured.

Table 3-2 headset_charge_current

Type	Name	Size	Description
unsigned	`charge`	1 bit	Enable or disable the charger
`power_boost_enable`	`boost`	2 bits	The boost setting. This can be: `power_boost_disabled` (0), `power_boost_internal_enabled` (1) or ■ `power_boost_external_enabled` (2)
`headset_vsel`	`vsel`	1 bit	`vsel_chg` (0) runs the chip from the charger supply; `vsel_bat` (1) runs the IC from the battery.
unsigned	`power_off`	1 bit	Powers off the Headset
unsigned	`disable_leds`	1 bit	Turns off all LEDs and disables further updates if set. Re-enables LEDs if not set.
unsigned	unused	2 bits	Unused

Table 3-3 describes charger termination settings.

Table 3-3 headset_charge_termination

Type	Name	Size	Description
`headset_vterm`	`type`	4 bits	`vterm_default` (0): does not modify termination voltage. `vterm_voltage` (1): sets the termination voltage to the level specified in `voltage` for use with BlueCore7 and later). `vterm_trim` (2): reduces the charger trim by the amount specified by `trim` for pre-BlueCore7 parts.
unsigned	`trim`	4 bits	The amount to reduce the trim if type is set to `vterm_trim`.
unsigned	`voltage`	8 bits	The voltage (i.e. mV/20) to set if type is set to `vterm_voltage`.

A headset_charge_setting structure, see Table 3-4, is configured for each threshold.

Table 3-4 headset_charge_setting

Type	Name	Size	Description
headset_charge_current	current	1 word	Current setting
headset_charge_termination	termination	1 word	Termination voltage setting

3.3 Configuration of headset power settings

The Headset power settings are stored in headset_power_settings, see Table 3-5.

Table 3-5 headset_power_settings

Type	Name	Size	Description
headset_battery_setting [6]	bat_events	6 words	Battery settings
headset_charge_setting [8]	chg_settings	16 words	Charger settings

The full Headset power configuration includes the power library configuration and is stored in PSKEY_USR0, see Table 3-6.

Table 3-6 PSKEY_USR configuration bit fields

Type	Name	Size	Description
power_config	config	23 words	Power library configuration
headset_power_settings	settings	20 words	Headset power manager configuration

3.4 Charger status

When the Headset receives a POWER_CHARGER_STATUS_IND message it generates an event based on the charger state, on which LED filters can be configured. Charger states map to the events in Table 3-7.

Table 3-7 Charger state to event mapping

Charger State	Event Generated
power_charger_trickle power_charger_fast power_charger_boost_internal power_charger_boost_external	EventChargeInProgress
power_charger_complete	EventChargeComplete
power_charger_disconnected	None
power_charger_disabled	EventChargeDisabled

4 USB power management

If the USB battery charging option is enabled in the device class bitmask, then the USB module also influences charger behavior, this option must be enabled if any other USB profiles are supported.

On BlueCore5 resistive detection is used to determine whether the Headset is attached to a standard host/hub or a dedicated charger.

On BlueCore7 voltage source detection is used and can also determine the difference between a dedicated charger and a charging host/hub.

Table 4-1 describes USB configuration.

Table 4-1 usb_config

Type	Name	Size	Description
`usb_device_class_type`	`device_class`	1 word	Bitmask of supported USB profiles
`headset_charge_current`	`i_disc`	1 word	Current setting when USB not attached
`headset_charge_current`	`i_susp`	1 word	Current setting when suspended
`Headset_charger_current`	`i_susp_db`	1 word	Current setting when suspended with dead battery prior to enumeration
`headset_charge_current`	`i_att`	1 word	Current setting when USB attached
`headset_charge_current`	`i_att_trickle`	1 word	Current setting when USB attached and trickle charging
`headset_charge_current`	`i_conn`	1 word	Current setting when enumerated
`headset_charge_current`	`i_conn_trickle`	1 word	Current setting when enumerated and trickle charging
`headset_charge_current`	`i_chg`	1 word	Current setting when connected to a charging host/hub
`headset_charge_current`	`i_dchg`	1 word	Current setting when connected to a dedicated charger
`headset_charge_current`	`i_lim`	1 word	Current setting when host current limiting detected

Table 4-1 usb_config (cont.)

Type	Name	Size	Description
unsigned	`audio_always_on`	1 bit	Route USB audio when the Headset is attached even if not in use
unsigned	`unused`	1 bit	Reserved
unsigned	`plugin_type`	2 bits	The plugin type to use for USB audio
unsigned	`plugin_index`	4 bits	The plugin index to use for USB audio
unsigned	`attach_timeout`	4 bits	The time after USB is attached, (in seconds), after which the Headset resets to low power bootmode
unsigned	`deconfigured_timeout`	4 bits	The time after USB is deconfigured, (in seconds), after which the Headset resets to low power bootmode

4.1 Combining current setting

USB current settings are applied in combination with any current limits imposed by the Headset Power Manager. When combining two current settings each field in the current setting is compared and the value that results in the lowest current draw from USB is applied, see Table 4-2.

Table 4-2 Current limit settings

Name	Combination
`charge`	Disables the charger if either setting is `FALSE`
`boost`	Disables the charger if either setting is `power_boost_disabled`
`vsel`	Headset draws power from VBAT if either setting is `vsel_bat`
`power_off`	Headset powers off if either setting is `TRUE`
`disable_leds`	Disables LED indications if either setting is `TRUE`
`current`	The lowest charge current setting is applied

4.2 Standard charger

When the Headset is attached to a charger but does not detect a USB source, the Headset applies the `i_disc` current setting. This is generally the case when a USB cable is used to connect the Headset to a power brick that does not support the USB battery charging specification.

4.3 Standard host/hub

When the Headset detects that it is attached to a standard USB host/hub it applies either the `i_att` or `i_att_trickle` current setting. The overall current draw prior to enumeration must not exceed 100 mA, which setting is applied depends on the state of the charger. When in fast charge the Headset draws the full charge current for the `i_att` setting, by default 100mA which requires the Headset draw power from the battery and not the USB supply. When in trickle charge the Headset draws much less than the full charge current so the `i_att_trickle` setting may select to draw power from the USB supply.

If the Headset detects that it has been suspended by the host, then `i_susp` is applied. This setting must limit the headset's current consumption to 2.5 mA averaged over 1 second, which by default is achieved by turning off the charger, disabling LEDs and selecting to run the Headset from the battery supply. The exception to this rule is if the battery voltage is below the critical threshold and the headset is suspended prior to enumeration, in which case `i_susp_db` is applied to ensure that the Headset is able to enumerate in order to charge the battery. By default this setting turns off the charger and disables LEDs, but allows the headset to draw power from the USB supply.

When the Headset is enumerated the Headset applies either the `i_conn` or `i_conn_trickle` current setting. The overall current draw once enumerated may be up to the maximum power consumption setting (which can be up to 500 mA and is by default). By default both these settings enable the charger with a charge current of 150 mA and draw power from the USB supply.

4.3.1 Failure to enumerate

If `attach_timeout` is set to a value other than zero a timeout message is sent when the Headset detects it has attached to a standard host/hub. Similarly if the `deconfigured` timeout is set to a value other than zero a timeout message is sent when the Headset is deconfigured by a standard host/hub. If the Headset has not enumerated before either of these timeouts fire the Headset resets into bootmode 3, i.e. USB Low Power mode. In most cases this is not necessary, however if the host cannot supply the requested maximum power setting the Headset must take action in order to ensure that enumeration succeeds.

By default the maximum power consumption is set to 100 mA in this boot mode, meaning the Headset must limit its current consumption to 100 mA even once enumerated. In this mode `i_conn` is identical to `i_att` and `i_conn_trickle` is identical to `i_att_trickle`. This ensures that the Headset can still enumerate if the host cannot supply the ideal charge current.

When the Headset detects the charger has been removed, it resets to the default bootmode to request the highest possible maximum power when enumerating again.

In both cases, provided the Headset panic recovery feature is enabled, which it is by default, the Headset restores to its previous state and re-establishes any Bluetooth connections that were present prior to resetting boot mode.

4.4 Charging downstream port

When the Headset detects, it is attached to a host/hub capable of operating as a charging downstream port it applies the `i_chg` current setting. Once attached the Headset is permitted to draw up to 1.5 mA from a charging downstream port at any time, even if suspended.

4.5 Dedicated charging port

When the Headset detects, it is attached to a dedicated charging port it applies the `i_chg` current setting. When attached the Headset is permitted to draw up to 1.5 mA from a charging downstream port at any time.

4.6 Current limiting

When drawing above 500 mA from a charging downstream port or dedicated charging port the Headset is required to monitor VCHG to ensure, it does not drop below 2.0 V. The charger hardware automatically switches off if VCHG drops below 3.3 V. If the power library detects a drop in VCHG before this happens, then a `POWER_CHARGER_VOLTAGE_IND` message is sent to the application indicating that V_{CHG} has dropped below the configured threshold. On receipt of this message `i_lim` is applied and should be set to draw 500 mA or less.

Document references

Document	Reference
Audio Sink Application User Guide	80-CT439-1/CS-00236868-UG

Terms and definitions

Term	Definition
ADC	Analog-to-Digital Converter
ADK	Audio or Application Development Kit
AIO	Asynchronous Input/Output
BlueCore	Group term for the range of QTIL Bluetooth wireless technology ICs
Bluetooth	Set of technologies providing audio and data transfer over short-range radio connections
CFM	Confirmation
IC	Integrated Circuit
IND	Indication
LED	Light Emitting Diode
PIO	Programmable Input Output
QTIL	Qualcomm Technologies International, Ltd.
USB	Universal Serial Bus