# TCCXXXX
# BATTERY DRIVER
# USER GUIDE

**TCCXXXX_BATTERY_DRIVER_USER_GUIDE**

**Rev. 0.10**

**Feb. 16. 2012**

Telechips

# DISCLAIMER

All information and data contained in this material are without any commitment, are not to be considered as an offer for conclusion of a contract, nor shall they be construed as to create any liability. Any new issue of this material invalidates previous issues. Product availability and delivery are exclusively subject to our respective order confirmation form; the same applies to orders based on development samples delivered. By this publication, Telechips, Inc. does not assume responsibility for patent infringements or other rights of third parties that may result from its use.

Further, Telechips, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.
No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of Telechips, Inc.
This product is designed for general purpose, and accordingly customer be responsible for all or any of intellectual property licenses required for actual application. Telechips, Inc. does not provide any indemnification for any intellectual properties owned by third party.
Telechips, Inc. can not ensure that this application is the proper and sufficient one for any other purposes but the one explicitly expressed herein. Telechips, Inc. is not responsible for any special, indirect, incidental or consequential damage or loss whatsoever resulting from the use of this application for other purposes.

# COPYRIGHT STATEMENT



# Important Notice

This product may include technology owned by Microsoft Corporation and in this case it cannot be used or distributed without a license from Microsoft Licensing, GP.

**For customers who use licensed Codec ICs and/or licensed codec firmware of mp3:**
"Supply of this product does not convey a license nor imply any right to distribute content created with this product in revenue-generating broadcast systems (terrestrial. Satellite, cable and/or other distribution channels), streaming applications(via internet, intranets and/or other networks), other content distribution systems(pay-audio or audio-on-demand applications and the like) or on physical media(compact discs, digital versatile discs, semiconductor chips, hard drives, memory cards and the like). An independent license for such use is required. For details, please visit http://mp3licensing.com".

**For customers who use other firmware of mp3:**
"Supply of this product does not convey a license under the relevant intellectual property of Thomson and/or Fraunhofer Gesellschaft nor imply any right to use this product in any finished end user or ready-to-use final product. An independent license for such use is required. For details, please visit http://mp3licensing.com".

**For customers who use Digital Wave DRA solution:**
"Supply of this implementation of DRA technology does not convey a license nor imply any right to this implementation in any finished end-user or ready-to-use terminal product. An independent license for such use is required."

**For customers who use DTS technology:**
"Supply of this implementation of DTS technology does not convey a license, exhaust DTS' rights in the implementation, or imply a right under any patent, or any other industrial or intellectual property right of DTS to use, offer for sale, sell, or import such implementation in any finished end-user or ready-to-use final product. Notice is hereby provided that a license from DTS is required prior to such use."

"This product made under license to U.S. Patents 5,451,942; 5,956,674; 5,974,380; 5,978,762; 6,487,535; 6,226,616 and/or foreign counterparts."

"© 1996 – 2010 DTS, Inc."

## Revision History

| Date | Version | Description |
|---|---|---|
| 2012-02-16 | 0.10 | This document is a guide to the Battery Driver. Initial release |
| | | |
| | | |
| | | |
| | | |

TABLE OF CONTENTS

**Contents**

1 Battery Driver ..... 1-1
2 Battery Driver source ..... 2-1
3 Battery Percentage ..... 3-1
4 ADC value array ..... 4-1
5 How to adjust adc value ..... 5-2
6 Compensation ..... 6-2
    6.1 Load compensation ..... 6-2
    6.2 AC compensation ..... 6-3
    6.3 Resume compensation ..... 6-3
7 Battery check loop time ..... 7-3

# 1 Battery Driver

- This guide shows how to adjust battery driver for actual device.

# 2 Battery Driver source

- Source name : tcc_battery.c
- You can find driver source at /kernel/drivers/power/tcc_battery.c

# 3 Battery Percentage

- Battery driver source code includes an array which is used to detect current battery level based on ADC value.
  - Percentage : Current battery level / Fully charged battery level x 100
  - High : maximum ADC values for the battery level
  - Low : minimum ADC value of the battery level

Battery level will be displayed by using percent value.
If percent is changed according to new adc value, battery level display will be changed.
If battery level is decreased and reaches 15 percent, low battery warring will be displayed.
And device will be turned off if battery level reaches 0 percent.

# 4 ADC value array

Example of array

```
const int m805_892x_vol[] = {
    3700,  // 4.2v
    3613,  // 4.1v
    3522,  // 4.0v
    3425,  // 3.9v
    3330,  // 3.8v
    3235,  // 3.7v
    3144,  // 3.6v
    3043,  // 3.5v
    3010,  // 3.45v
};
```

- By adjusting these values, you can
  - change the level of battery indication for a battery level and also decide when low battery warring is displayed and the device should be turned off.
  - decide timing when low battery warning is diplayed.
  - decide timing when the device is turned off

# 5 How to adjust adc value

- To help you find ADC value for a battery level, ADC values can be displayed through console.
- To activate this monitoring function, you have to change the feature TRACE_BAT from 0 to 1 as below.

**tcc_battery.c**
**#define TRACE_BAT 1 // activate adc output**
**#define TRACE_BAT 0 // deactivate adc output**

- Following figure shows an example of ADC output

**[BATT] batt id=0,*voltage=653*, temp=100, current=100, level=100%, charging source=1, charging_enable=1, full=100**
**[BATT] batt id=0,*voltage=654*, temp=100, current=100, level=100%, charging source=1, charging_enable=1, full=100**
**[BATT] batt id=0,*voltage=654*, temp=100, current=100, level=100%, charging source=1, charging_enable=1, full=100**

By using this output, you can find the ranage of ADC value for current battery level and modify the array to hold correct ADC value for actual hardware.

# 6 Compensation

If you want to compensate adc value, you can change compensation value

#define COMPENSATION_LOAD 1
#define COMPENSATION_AC 1
#define COMPENSATION_RESUME 1

1 -> Compensation on , 0 -> Compensation off

## 6.1 Load compensation

This compensation is for usage of CPU(AP)
If load of cpu is increased, voltage of battery is dropped.

```
if(1) { //adcValue < pbattery_vols[6]){
        if(clk_get_rate(cpu_clk) > 800000000)
                adcValue += 30;
        else if (clk_get_rate(cpu_clk) > 700000000 || clk_get_rate(ddi_clk) > 360000000)
                adcValue += 25;
        else if (clk_get_rate(cpu_clk) > 600000000 || clk_get_rate(ddi_clk) > 340000000)
                adcValue += 20;
        else if (clk_get_rate(cpu_clk) > 500000000 || clk_get_rate(ddi_clk) > 310000000)
                adcValue += 10;
        else if (clk_get_rate(cpu_clk) > 400000000 || clk_get_rate(ddi_clk) > 260000000)
                adcValue += 5;
}
```

So, ADC value needs compensation by load.
You should change these values according to device.

### 6.2 AC compensation

```
#if defined(CONFIG_REGULATOR_AXP192)
                    if(axp192_charge_status() != 0x02){
#endif
                    if(adcValue < pbattery_vols[5])
                            adcValue -= ADJUSTCHARGE1;
                    else if(adcValue >= pbattery_vols[5] && adcValue < pbattery_vols[0])
                            adcValue -= ADJUSTCHARGE2;
                    else if(adcValue >= pbattery_vols[0])
                            adcValue -= (ADJUSTCHARGE3);
#if defined(CONFIG_REGULATOR_AXP192)
                    }
#endif
```

When AC power is plugged, voltage of battery is increased.
So, adcValue must be decreased as much as AC power.
You should change these values according to device.

### 6.3 Resume compensation

```
    if(!g_ac_plugin){
            adjustResume = 80;
            if(g_usb_online)
                    adjustResume = 60;
    }
    else
            adjustResume = 20;

    if(in_suspend)
            adcValue -= adjustResume;
```

When device is waken up, adcValue is changed.
So, you should adjust adcValue above value in red according to device.

## 7 Battery check loop time

Basically, ac power detection loop time is 1500ms.And basic time unit is same.

```
#define BATT_CHECK_LOOP 1500 //ms
```

Reading adc value loop time is BATT_CHECK_LOOP X times.

For example

batt_check_times = 6;

➔ Reading adc value loop time = 1500ms X 6 => 9s

You can change this value according to device.