Application Report
SLUA364–November 2005

Theory and Implementation of Impedance Track™ Battery Fuel-Gauging Algorithm in bq20z8x Product Family

Yevgen Barsukov *PMP Portable Power*

ABSTRACT

In fuel gauge solutions, current integration and voltage correlation algorithms suffer from a decrease in accuracy with battery aging and also require extensive data collection. This application report outlines the theory of Impedance Track™ (IT) technology that overcomes these problems. Implementing the IT algorithm in the Texas Instruments bq20z8x family is reviewed and setting the data flash constants associated with the fuel-gauging algorithm is described in detail.

Contents

List of Figures

1 Summary of the Algorithm Operation

The Impedance Track™ gas gauge algorithm(1) uses three types of information to calculate remaining capacity (*SBS.RemainingCapacity( )*) and full charge capacity (*SBS.FullChargeCapacity( )*).

1. Chemical: depth of discharge (DOD), and total chemical capacity Q_{max}
2. Electrical: internal battery resistance dependence on DOD
3. External: load, temperature

SBS.FullChargeCapacity( ) is defined as the amount of charge passed from a fully charged state until the voltage defined in *DF:Terminate Voltage* flash constant is reached at a given rate of discharge, after subtracting the reserve capacity (*DF:Reserve Capacity*).

Note that it depends on the rate of discharge and is lower at higher rates and low temperatures because the cell I x R drop causes the Terminate Voltage threshold to be reached earlier.

(1) Impedance Track algorithm is protected by US Patents US6832171, US6789026, and US6892148.

Impedance Track is a trademark of Texas Instruments.

2 Detailed Description of Parameters Updated by the Gas Gauge Algorithm

2.1 Modes of Algorithm Operation

The algorithm differentiates between *charge*, *discharge*, and *relaxation* modes of operation. During *charge* mode, the SBS.OperationStatus( ) [DSG] bit is cleared, and during *discharge* and *relaxation* mode, it is set. Entry and exit of each mode is controlled by Data Flash (DF) parameters in the subclass Gas Gauging: Current Thresholds section as illustrated in Figure 1. Charge mode is exited, relaxation mode is entered when *SBS.Current( )* goes below *DF:Quit Current* and after a *DF:Chg Relax Time* period. Discharge mode is entered when *SBS.Current( )* goes below *DF:Dsg Current Threshold*. Discharge mode is exited, relaxation mode is entered when *SBS.Current( )* goes above —*DF:Quit Current* threshold and after a *DF:Dsg Relax Time* period. Charge mode is entered when *SBS.Current( )* goes above *DF:Chg Current Threshold*.

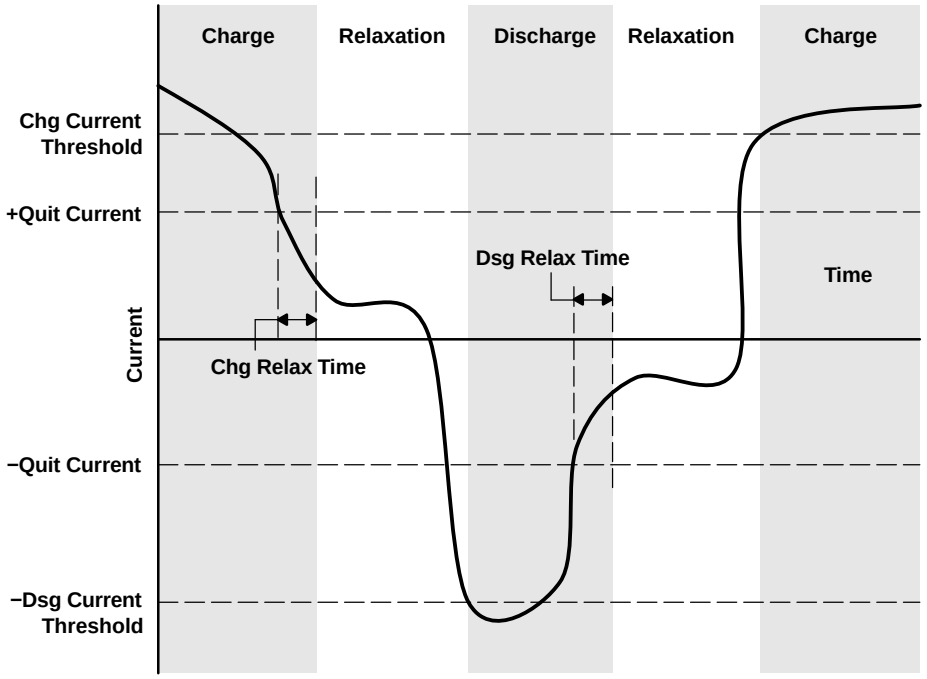

Figure 1. Example of the Algorithm Operation Mode Changes With Varying *SBS.Current( )*

2.2 Update of Chemical Depth of Discharge (DOD)

The gas gauge updates information on the chemical depth of discharge (DOD_0) based on open-circuit voltage (OCV) readings when in a relaxed state. This is done for each cell separately. DOD is found by correlating DOD with OCV using a predefined table DOD(OCV,T) stored as reserved data flash parameters. The table is specific for a particular chemistry such as $LiCoO_2$/carbon (default settings), LiMn2O4/carbon, etc., and can be identified by reading the chemistry ID through sending SBS.ManufacturerAccess( ) command 0008, then reading SBS.ManufacturerAccess().The gas gauge can be set up for a particular chemistry by using the relevant firmware file (*.senc) that can be downloaded from the bq20zXX production folder on power.ti.com.

Figure 2 shows the timing of parameter updates during the relaxation mode. First, OCV readings and DOD_0 calculations are taken after a 30-minute relaxation period has passed. Then, OCV readings continue to be taken every 100 seconds. DOD is calculated based on each measured OCV reading using the linear interpolation DOD = f(OCV,T). Integrated PassedCharge is set to zero at each DOD_0 update.

If the current during the OCV reading is non-zero, then an IR correction is done. The first iteration of DOD is found from the uncorrected OCV reading; then the resistance value is found from the R(DOD) table and used to correct the OCV value as OCV′= OCV - I x R. Then, the corrected DOD is found from OCV′. This method achieves the best accuracy if the current during relaxation mode is below the C/20 rate. This is why it is recommended that the DF.Quit Current not exceed C/20.

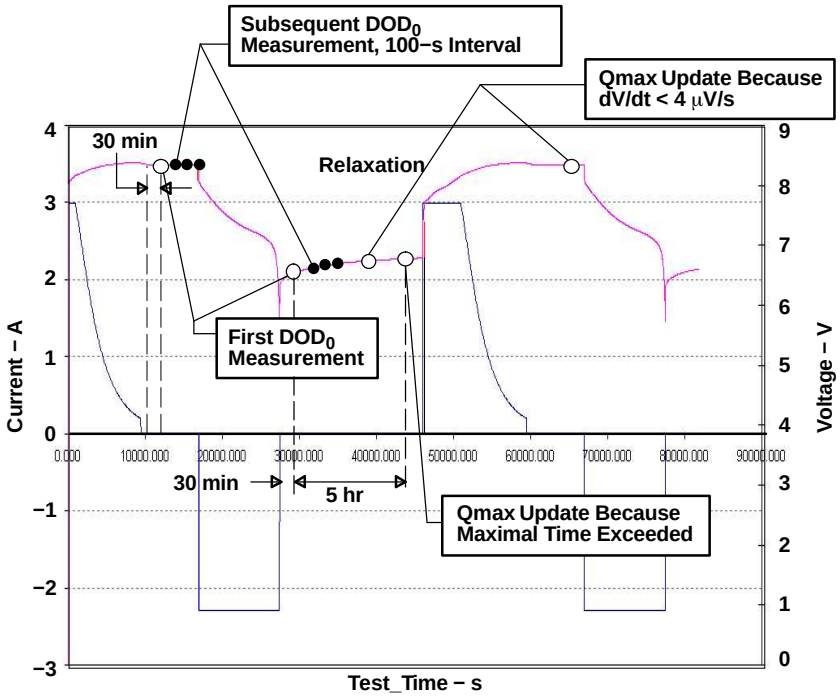

Figure 2. Timing of DOD_0 and Qmax Updates During Relaxation Mode

If no DOD_0 has been measured until the relaxation state is exited, the previous DOD_0 is used along with the PassedCharge integrated since the last DOD reading.

During charge and discharge modes, the *present* DOD is recalculated every second as DOD = DOD_0 + PassedCharge/Qmax. DOD is used for determining when a resistance update needs to occur, as well as the starting point for a Remaining Capacity (and FCC) calculation. Remaining capacity calculations occur immediately after discharge onset, at every resistance update, and after entering relaxation mode.

2.3 Update of Qmax

Maximal chemical capacity Q_{max} for each serial cell is stored in Data Flash as *DF:QmaxX*, where X=0, 1, 2, or 3, the cell serial number.

The GG updates Q_{max} based on two DOD readings made before and after charge or discharge, for each serial cell separately. For example, DOD_1 is taken during relaxation, then discharge mode starts, and PassedCharge is integrated. Following this, another relaxation mode is entered, and DOD_2 is taken.

DOD1 and DOD2 are calculated from OCV readings in a well-relaxed state, as exemplified in Figure 2. A well-relaxed state is detected if dV/dt < 4 μV/s or maximal waiting time of 5 hr is exceeded. The first condition is satisfied in typical batteries after about 1 hr if DOD is between 0% and 80%, and 3–4 hr if DOD is above 80%. At a low temperature, relaxation takes a longer time.

In order to ensure high accuracy of DOD measurement, Qmax calculation do not occur if the temperature is above 40°C or below 10°C. It also does not occur if at least one of voltage measurements for DOD_1 or DOD_2 was taken in the cell voltage range between 3737 mV and 3800 mV because of flat OCV(DOD) dependence in this range. These limits are chemistry dependent and will be specified separately for different chemistries.

Qmax is calculated as Qmax = PassedCharge / (DOD_2– DOD_1).

The data flash constant *DF.Update Status* increments by 1 when the first Qmax update takes place (e.g., from 4 to 5 if no resistance updates were made, or from 5 to 6 if a resistance update was made). *SBS.Max Error( )* becomes 5% in the first case and 1% in the second case

PassedCharge has to be more than 37% of *DF:Design Capacity* for an update to occur. For the first cycle (with *DF:Update Status* = 4), 90% of *DF:Design Capacity* is required because this cycle takes place in the factory settings and Qmax is learned for the first time.

In order to prevent Qmax fluctuations, a first-order smoothing filter is applied to all Qmax readings except in the first cycle. Readings with lower PassedCharge are assigned lower weights in the smoothing.

2.4 Update of Resistance

Resistance is updated during discharge, as summarized in Figure 3. The first resistance update happens after 500 seconds, to prevent distortion from transients after load onset.

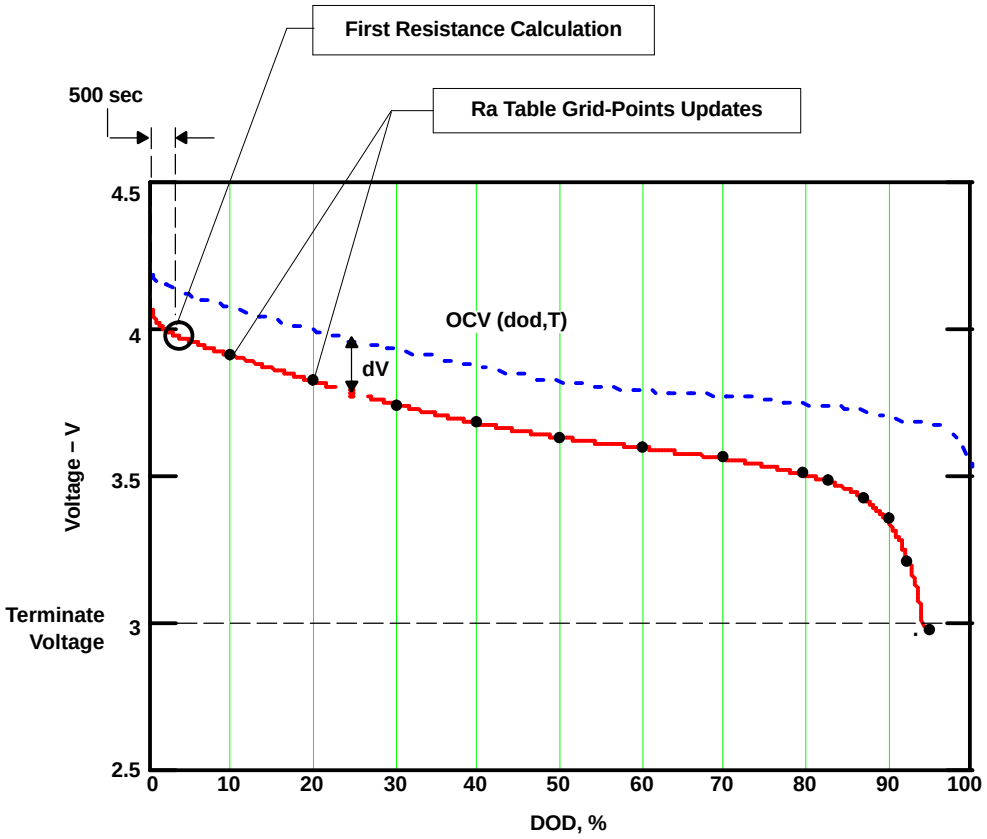

Figure 3. Impedance Updates

Calculation is performed by comparing the measured voltage with the OCV value at the same DOD, that is taken from the OCV(DOD,T) table:

dV = V – OCV(DOD,T)

R(DOD) = dV/I

Resistance measurements are taken continuously and stored in RAM.

Resistance is updated in the Data Flash (in *DF:Ra Table*) after each 10% of DOD charge is exceeded (DOD charge is PassedCharge/Qmax). When DOD reaches 80%, resistance is updated after each 3.3%. The final resistance update is made after discharge is terminated.

The constant *DF.Update Status* increments by 1 when the first grid-point resistance update takes place (e.g., from 4 to 5 if no Qmax updates were made before, or from 5 to 6 if Qmax updates were made before).

Before storage to Data Flash, resistance values are normalized to 0°C as Ra[dod] = R[dod]/exp(Rb[dod] x T) where R is the measured resistance value at a given DOD, Rb[DOD] is the value of temperature coefficient of impedance change at a given DOD stored as a reserved data-flash table, and T is temperature in °C. Note that values of resistance normalized to 0°C are somewhat larger than values at room temperature and so cannot be directly compared with R=dV/I values.

Resistance values for the grid points with a higher DOD than presently updated are scaled by the same factor as the present grid-point change, e.g., by factor Ra_new/Ra_old. In this way, faster convergence of the resistance profile is achieved.

Values in *DF.Ra Table* are stored in mΩ units, in the format CellX Ra N where X is a cell serial number from 0 to 3, and N is grid-point number from 0 to 14 that corresponds to 10% increments of DOD until 80%, and then 3.3% increments of DOD. The CellX flag and xCellX flag are used for interchanging the data-flash column usage for reducing the number of DF writes. A flag value of 55 indicates the presently used data column; 00 indicates the presently unused data column.

If during resistance update, the DOD exceeds 100%, or resistance appears negative, which are both indications of a too-small DF.QmaxX initial guess, DF.QmaxX increments by 10% and all resistances are recalculated. This is normal behavior during the first *learning* cycle. However, if the initial guess of DF.QmaxX was too far from the correct value, the second cycle might be needed to achieve full resistance accuracy. To avoid this, set DF.QmaxX to a value specified by the cell manufacturer, multiplied by the number of parallel cells.

2.5 Update of Temperature Model

Because temperature changes significantly during the course of a discharge, the algorithm needs to be able to predict the future temperature. This is needed for temperature correction of battery impedance ® = Ra x exp(Rb x T)) during voltage simulation near the end of a discharge. To achieve this, the algorithm collects T(t)-dependence data during discharge. It is used to update parameters of a simple thermal model including a heat exchange coefficient and a thermal time constant. These parameters are updated at the same time as resistances. The algorithm also records the outside temperature (T_out) during relaxation periods. These parameters are used to define a function T(t, T_start) that calculates a temperature profile starting from present temperature, T_start, and continuing until the end of discharge.

2.6 Update of Remaining Capacity (RM) and Full Charge Capacity (FCC)

Update of RM and FCC takes place after each resistance grid-point update, at the end of discharge, and at the exit of relaxation mode.

FCC consists of 3 parts:

SBS.FullChargeCapacity( ) = Q_{start} + PassedCharge + RM

Components of FCC are indicated in an example in Figure 4.

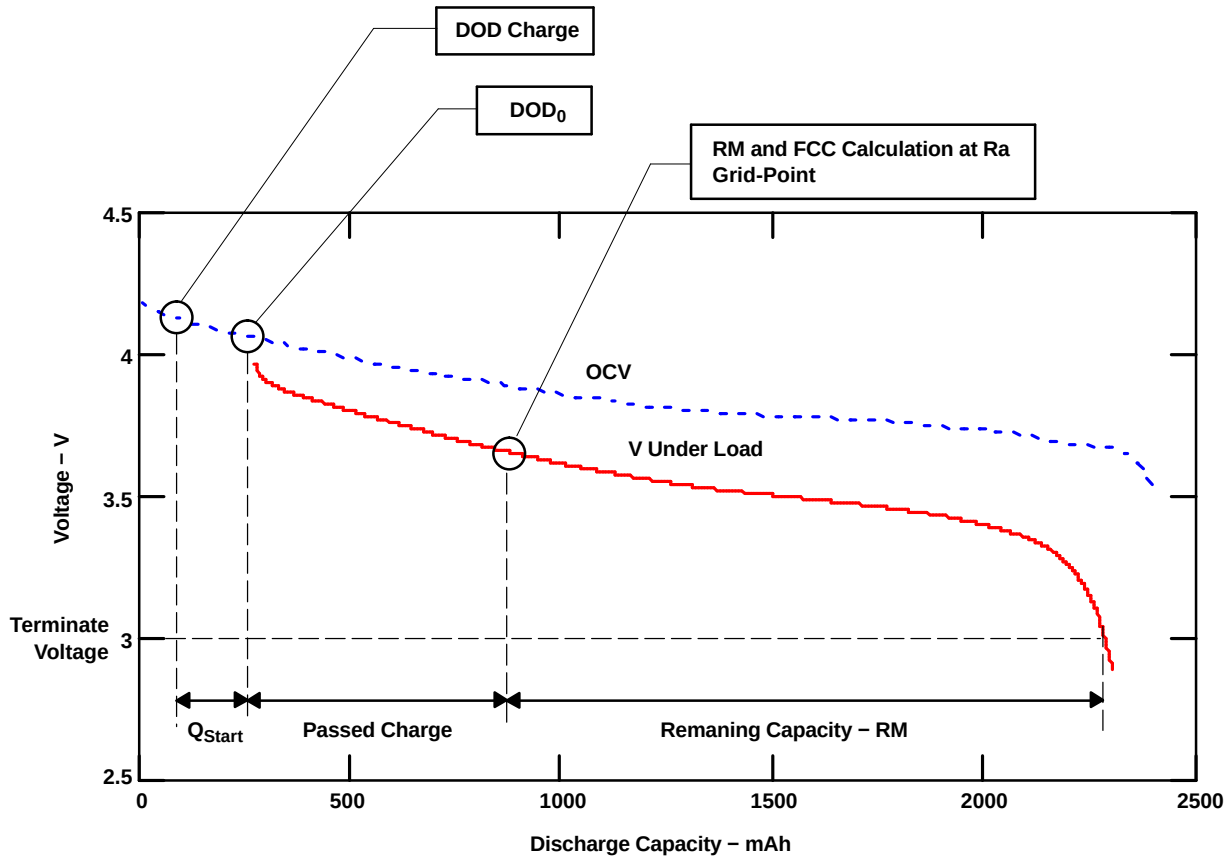

Figure 4. Components of SBS.FullChargeCapacity( ) Value

1. Q_{start} is the charge that would have passed to make DOD = DOD_0 from a fully charged state (DODcharge). For a fully charged battery, Q_{start}=0. Q_{start} is recalculated at the exit of the relaxation mode. In the case of constant current, it is simply Q_{start} = Q_{max}× (DOD0 – DODcharge), but for the constant power case a voltage simulation is run. DODcharge is assigned equal to DOD_0 at first DOD_0 update, after charge termination by taper current. Note that DODcharge is somewhat higher than 0 because chargers typically do not charge a battery to full.
2. PassedCharge is the coulomb count integrated during the present discharge or charge, and set to zero at every DOD_0 update.
3. Remaining capacity is calculated after each resistance grid-point update and at the end of discharge.

SBS.RemainingCapacity( ) (RM) is calculated using a voltage simulation. The GG starts a simulation at the present DODstart = DOD_0 + PassedCharge/Qmax and continue calculating voltage V(DODx,T) = OCV(DODx,T) + I x R(DODx,T) by incrementing DOD with dDOD increment of 4%. DOD[i]=DODstart + dDOD x I. This incrementing is continued until the simulated voltage V(DOD[i]) becomes less than *DF.Terminate Voltage*. Once that happens, the final DOD *SBS.RemainingCapacity( )* = (DODfin-DODstart) x Q_{max} is detected. Note that Q_{max} for the lowest capacity cell is used.

Current that is used in the simulation is the average current during the present discharge (several types of averaging can be selected using *DF:Load Select* data flash constant). A simulation can run in constant current mode (*DF:Load Mode* = 0) or constant power mode (*DF:Load Mode* = 1).

2.7 Update of SBS.Remaining Capacity( ) and SBS.RemainingStateOfCharge( ) Values

Although *SBS.FullChargeCapacity( )* is only updated at a few points during a discharge as previously described, the *SBS.RemainingCapacity( )* is updated continuously (every 1 second) based on the integrated charge. *SBS.RemainingCapacity( )* = RM – Q_integrated where Q_integrated is charge passed since the last RM calculation. The value of *SBS.RemainingCapacity( )* is also used to update *SBS.RelativeStateOfCharge( )* every second as *SBS.RelativeStateOfCharge( )* = *SBS.RemainingCapacity( )* x 100/*SBS.FullChargeCapacity( )*.

The same value is used to calculate the run-time to empty as *SBS.RunTimeToEmpty( )* = *SBS.RemainingCapacity( )* / *SBS.AverageCurrent( )*.

Note that even if a simulation of RM is run in constant power mode (*DF:Load Mode* = 1), the reporting of *SBS.RemainingCapacity( )* and *SBS.RemainingRunTime( )* can be done either based on mAh or in 10mWh values. The mAh or mWh reporting depends on the setting of *SBS.BatteryMode( ) [CAPACITY_MODE]* bit (0=mAh, 1=10mWh). In case of a second setting, the run-time-to-empty is calculated as SBS.RunTimeToEmpty( ) = SBS.RemainingEnergy( ) / SBS.AveragePower( ) and is generally more accurate for most devices because of increased power consumption at low voltages.

2.8 Update of SBS.Max Error()

SBS.Max Error() is an estimate of maximal error of SBS.RSOC. Initially, it is set to 100% because the fuel gauge has no information about the battery. After Qmax is learned, or resistance is learned (as indicated by DF.Update Status = 5), SBS.Max Error changes to 5%. After the second part of the database (resistance or Qmax) is learned (Update Status = 6), SBS.Max Error changes to 1%. SBS.Max Error( ) is increased to reflect the number of cycles since the last Qmax. This is achieved by storing the cycle number when Qmax was last updated in an internal variable Qmax_cycle. SBS.Max Error is then calculated as Max Error + (SBS.Cycle Count() – Qmax_cycle) x 0.05. This means that SBS.Max Error increases by 1% in 20 cycles, e.g., only occasionally is a Qmax update needed to maintain high accuracy.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265