Application Report
SBAA155A–July 2007–Revised September 2007

Reducing Analog Input Noise in Touch Screen Systems

Wendy Fang — *HPA Precision Analog Applications*

ABSTRACT

This application report discusses some common methods for reducing noise in the analog input circuitry of a resistive touch screen controller. In terms of hardware design, the printed circuit board layout and grounding are the keys; the additional noise-decoupling capacitor in a touch screen controller input pin may also help. On the software side, several simple, widely-applied filtering algorithms are presented as solutions for noise reduction.

Contents

List of Figures

SPI is a trademark of Motorola, Inc.
I^2C is a trademark of NXP Semiconductors.
All other trademarks are the property of their respective owners.

1 Introduction

In practical applications, problems associated with noise in electronic devices always seem unavoidable. Noise problems are especially common and significant in human and machine interface systems, such as a touch screen controller (TSC). Figure 1 illustrates a typical TSC system.

Figure 1. Touch Screen System with TI's ADS7846 Touch Screen Controller

In a TSC system, the touch panel is usually assembled on top of an LCD display, and therefore, excessive noise from both the display and back-light is easily coupled into the touch panel. Additionally, as the human and machine (computer) interface device, the panel is designed for users to touch. Touching the panel generates noise and parasitics because of panel mechanical vibrations and transients, as well as a result of electrostatic discharge (ESD) and electromagnetic pulses (EMP) from users and the operating environments.

Such noise is often picked up by the TSC device from its analog input circuitry, as shown in Figure 2. This noise can drastically reduce the accuracy and reliability of the entire touch screen system.

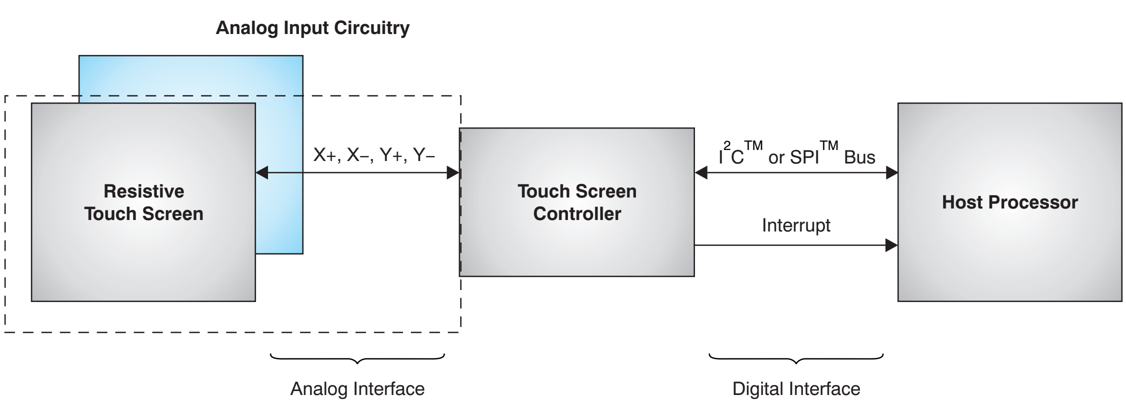


Figure 2. Block Diagram of a Typical 4-Wire Resistive Touch Screen System

This application report presents the most common methods to reduce noise at the TSC input circuitry, including:

- Attention to printed circuit board (PCB) layout;
- Additional de-coupling capacitors;
- Implementing averaging and filters; and
- Utilizing touch screen panel pressure parameters.

2 Layout

Good PCB layout practice can optimize TSC system performance, in addition to easing other design restrictions, reducing design/debug costs, and shortening product development time. Follow acceptable design practices for PCB layout in general.

In the specific case of a resistive touch screen, additional care should be given to the connection between the TSC and the touch screen; that is, the analog Interface as shown in Figure 2. Because resistive touch screens have fairly low resistance (100 Ω to 2000 Ω), the interconnection should be as short and robust as possible. Loose connections can be a source of error when the contact resistance changes with flexing or vibrations.

The analog-to-digital converter (ADC) in most resistive TSC devices is a successive-approximation response (SAR) ADC. The generic SAR architecture is sensitive to glitches or sudden changes in the power supply, ground connections, and digital inputs that occur just before latching the output of the analog comparator. Therefore, during any single conversion for an *n*-bit SAR converter, there are *n* windows in which large external transient voltages can easily affect the conversion result. Such glitches might originate from switching power supplies, nearby digital logic, and high-power devices. The degree of error in the digital output depends on the reference voltage, layout, and the exact timing of the external event. With this consideration in mind, power to the TSC should be clean and well-bypassed, with bypass capacitors between power and ground. These bypass capacitors must be placed as close to the TSC device as possible.

All ground pins of the TSC device should be connected to a clean ground point. In many cases, this point is the analog ground. Avoid connections that are too near the grounding point of a microcontroller or digital signal processor. If needed, run a ground trace directly from the converter to the power-supply entry or battery connection point. The ideal layout includes an analog ground plane dedicated to the converter and associated analog circuitry.

Many TSC applications have conflicting requirements for power, cost, size, and weight; each situation is unique. These suggestions are commonly-accepted PCB design practices for analog applications. Individual users should consult the specific layout requirements for the respective TSC device.

3 Noise Decoupling Capacitors

As noted earlier, noise can be a major source of error in touch screen applications (for example, applications that require a back-lit LCD panel). This electromagnetic inference (EMI) noise can be coupled through the LCD panel to the touch screen and cause flickering of the converted ADC data.

To reduce this type of error, using a touch screen with a bottom-side metal layer connected to ground is recommended. This configuration couples most of the noise to ground, and has been shown to be very helpful in many TSC applications.

Additionally, adding several decoupling capacitors, from Y+, Y–, X+, and X– to ground, is another way to reduce analog input noise. Note, however, that using these capacitors increases screen settling time and requires longer panel voltage stabilization times, and possibly longer pre-charge and sense times. See Ref. 12 for a discussion of these timing functions and features.

Figure 3 illustrates the outcomes of using a decoupling capacitor in the TSC input lines. Users may need to examine the relationship between the touch panel resistance, input circuit capacitance, and the TSC sample rate for possible decoupling solutions.

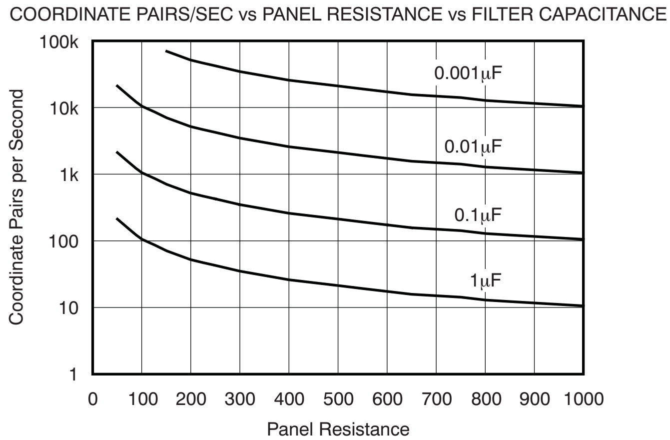


Figure 3. Touch Screen Controller Must Reduce Sample Rate While Input Capacitance and Panel Resistance Increase

The graph in Figure 3 indicates that a TSC device can produces more than 50,000 X/Y coordinate-pairs per second on a 300-Ω touch panel when the input capacitance is less than 0.001 µF. The same system with 1-µF capacitors can only produce 50 coordinate-pairs per second. Thus, in a real-world application, it is suggested to use a noise decoupling capacitor only as the last option for resolving TSC noise problems. *Add the capacitors only when absolutely necessary*.

When the capacitors must be used, first start with adding a small capacitor (for example, 0.01 µF), placed only on the negative (X– or Y–) pin.

4 Simple Software Average/Filter Algorithms

Another widely-used method for reducing noise in TSC applications is to average or filter multiple samples for each touch coordinate.

There are two possible places to implement the average/filter (see the TSC diagram in Figure 2). The first place is at the host processor, in the host software; the other (and a better) location is at the TSC, in the TSC built-in hardware (Ref. 1, Ref. 2). The second location is superior because of its advantage in reducing both digital interface traffic and host processor overhead.

When noise becomes one of the concerns in a given design, consider selecting and using a TSC device with a built-in filtering function, or develop noise reduction software routines to average/filter noise running in the host software.

Generally speaking, the data averaging or filtering can be expressed as a finite impulse response (FIR) filter, as shown in Equation 1:

$$X(k) = \sum_{n=0}^{N-1} b_n * x(k-n) \tag{1}$$

Where:

- X(k) is the refined touch data
- x(k) is the raw touch data
- b_n is the coefficient or weight factor of the filter
- N is the number of the multiple samples for each touch coordinate (N–1 is the order of the filter)

This section presents several simple average/filter algorithms, used by the host processor. These filters are very low-order because higher *N* costs higher overhead for the host processor, and creates higher analog and digital traffic as well as higher power consumptions.

4.1 Average

Averaging can be considered as Equation 1 with Equation 2:

$$b_n = \frac{1}{N} \tag{2}$$

Equation 3 shows this example: sample 4 X data, N = 4 and b_n = 1/N, and get the average:

$$X(k) = \sum_{n=0}^{3} x(k-n) / 4 = \frac{[x(k) + x(k-1) + x(k-2) + x(k-3)]}{4} \tag{3}$$

The software routine for this mathematical process is shown in Example 1.

Example 1. Averaging

```
//*********************************************************************************//
#define  N  4                         //  multiple sampling data N=4 times
unsigned int X;                       //  the refined/averaged X
unsigned int x[N];                    //  the 4 multiple samples raw touch data
byte i;                               //  a loop counter

// store the 4 raw sample data into x[4] array before run the following code

//
//************ Find the average value X of N samples in x[4] array ************//
//
X=x[0];                               // sum the N raw data
for (i=1;i<N;i++) { X += x[i]; }
X = X >> 2;                           // divide by N (here N=4)
//*********************************************************************************//
```

4.2 Weighted Average

In a touch screen system, there is usually more noise when the touch driver first powers on. Therefore, when the host samples a coordinate (for example, X), the driver should be programmed to stay on during the multiple samplings and to be turned off after the last data sampling. The first few data samples may be more noisy because the driver has just powered on. A weighted average can work better here; use Equation 4 in Equation 1:

$$b_n = \begin{cases} 0 & \text{for the first one or more data } x(k) \\ 1/M & \text{for the later } x(k) \end{cases} \tag{4}$$

Where:

- M is the number of averaged multiple data
- M < N

For example, sample 4 X data, N = 4; and discard the first two samples, then M = 2.

The software routine for this process is shown in Example 2.

Example 2. Weighted Averaging

```
//*******************************************************************************//
#define  N  4                        //  multiple sampling data N=4 times
unsigned int X;                      //  the refined/averaged X
unsigned int x[N];                   //  the 4 multiple samples raw touch data
byte i;                              //  a loop counter

// store the 4 raw sample data into x[4] array before run the following code

//
//******* Find the weighted average value X of N samples in x[4] array *******//
//
X=x[2];                              // sum the M raw data
for (i=2;i<N;i++) { X += x[i]; }
X = X >> 1;                          // divide by M (here M=2)
//*******************************************************************************//
```

4.3 Middle Value

The middle value of multiple samples can be considered as Equation 1 with Equation 5:

$$b_n = \begin{cases} 1 & \text{for middle value } x(k) \\ 0 & \text{for all other } x(k) \end{cases} \tag{5}$$

Obviously, a middle value is also a specific *weighted* average. The algorithm for finding the middle value (for example, at *N* = 3) is:

- Sample three touch data points; x(k), x(k–1), and x(k–2);
- Compare x(k) and x(k–1), and then compare the larger one (xH) to x(k–2);
- If xH is smaller than x(k–2), then X = xH and the calculation is done.
- Else: compare x(k–2) and non xH (that is, xL) ; the larger one is X, and the calculation is done.

Example 3 shows a software routine for this calculation.

Example 3. Middle Value Averaging

```
//*******************************************************************************//
unsigned int X;              //  the middle value X
unsigned int x[3];           //  the 3 multiple samples raw touch data

// store the 3 raw sample data into x[3] array before run the following code

//
//******** Find the middle value X among 3 samples x[0]. x[1] and x[2] ********//
//
if (x[0] >= x[1]) {          // compare x(k) and x(k-1)
xH=x[0];
xL=x[1];
} else {
xH=x[1];
xL=x[0];
}
if (xH <= x[2]) {            // larger one xH compare to x(k-2);
X=xH;                        // if xH smaller than x(k-2):  then X = xH
} else {                     // else: compare x(k-2) and non xH (i.e. xL)
if (x[2] >= xL) {            // the larger one is X
X=x[2];
} else {
X=xL;
}
}
//*******************************************************************************//
```

4.4 Averaging the Closest Data

There is another common average algorithm that uses only the data points that are close to each other, discarding any outlying data.

For example, Take X data three times (again, for *N* = 3), and use only the two points that are close to each other. It can be considered as Equation 1 with Equation 6:

$$b_n = \begin{cases} 1/2 & \text{for 2x data with values closer to each other} \\ 0 & \text{for 1x with values not close to each other (outlying)} \end{cases} \quad (6)$$

The example algorithm is:

- Sample three touch data, say x(k), x(k–1), and x(k–2);
- Get their absolute differentials: $d_1 = |x(k) – x(k-1)|$, $d_2 = |x(k–1) – x(k–2)|$ and $d_3 = |x(k–2) – x(k)|$,
- Find the smallest among d_1, d_2, and d_3

Example 4 gives the software routine for this calculation.

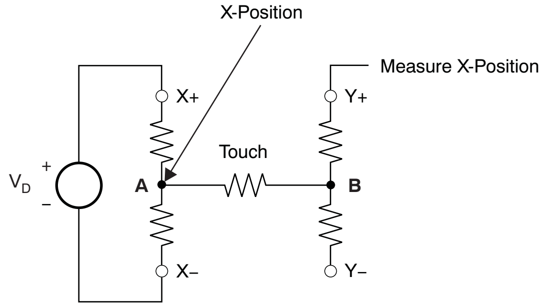
Example 4. Averaging Closest Data

```
//*********************************************************************************//
unsigned int X;              // the middle value X
unsigned int x[3];           // the 3 multiple samples raw touch data
int d[3];                    // the differentials of raw touch data
byte i;                      // a loop counter

// store the 3 raw sample data into x[3] array before run the following code

//
//*** Find the average value X of 2 close-valued samples in x[0]. x[1] and x[2] ***//
//
d[0] = x[0] – x[1];          // get the differentials
d[1] = x[1] – x[2];
d[2] = x[2] – x[0];
for (i=0;i<3;i++) {          // get the absolute differentials
if (d[i] < 0) {d(i) =  – d(i);}
}
if (d[0] < d[1]) {           // when d[0] < d[1]
if (d[0] < d[2]) {           // compare d[0] to d[2]
X = x[0] + x[1];             // d[0] is the smallest if d[0] < d[2] also
} else {
X = x[0] + x[2];             // d[2] is the smallest if d[2] <= d[0] < d[1]
}
} else {                     // otherwise, i.e.: when d[1] <= d[0]
if (d[1] < d[2]) {           // compare d[1] to d[2]
X = x[1] + x[2];             // d[1] is the smallest if d[1] < d[2] also
} else {
X = x[0] + x[2];             // d[2] is the smallest if d[2] <= d[1] <= d[0]
}
}
X >>= 1;
//*********************************************************************************//
```

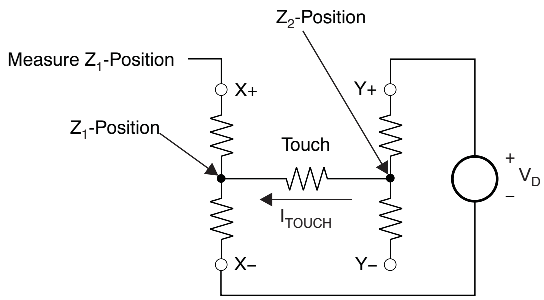
5 Touch Screen Panel Pressure

Another way to reduce noise and eliminate unreliable data, using software, is to take advantage of the TSC device pressure measurement. As we know, the host does not want the touch data set when the panel has not been touched (no pressure), or when it has not been firmly touched (very light pressure). Light touch or pressure on the panel may be simply understood by the system as false contacts because of panel mechanical vibration or other environmental factors.

The resistive touch panel pressure is proportional to the resistance between points A and B in Figure 4, denoted by R_Z , and Equation 7.



a) Add panel driving power to X+/X– pins, and read X-position from Y+ pin.



b) Add panel driving power to Y+/X– pins, and read Z_1-position from X+ pin and Z_2-position from Y– pin.

Figure 4. Pressure Measurement of Resistive Touch Panel

$$R_Z = \frac{(V_B - V_A)}{I_{TOUCH}} \tag{7}$$

Where:

- $V_A = V_D \times Z_1/Q$
- $V_B = V_D \times Z_2/Q$

Z_1 and Z_2 are the measured Z_1-Position and Z_2-Position, respectively; and Q corresponds to the TSC resolution. For example, Q = 256 if the TSC resolution is 8-bit, Q = 1024 if it is 10-bit, or Q = 4096 if it is 12-bit. When the touch driver or power (V_{DD}) is added between Y+ and X–, as shown in Figure 4b, $I_{TOUCH} = V_A / R_{XA} = (V_D \times Z_1/Q) / R_{XA}$, with the resistance at point A being $R_{XA} = R_X \times X / Q$, where X is the measured X-position when the X driver is ON, as shown in Figure 4a.. Thus, by bringing the expression of I_{TOUCH} here into Equation 7, we get Equation 8:

$$R_Z = \frac{V_D \times (Z_2 - Z_1) / Q}{V_D \times Z_1 / Q} \times R_{XA} = R_X \times X / Q \left(\frac{Z_2}{Z_1} - 1 \right) \tag{8}$$

Equation 8 is the first data sheet equation for many TI TSC devices; see Ref. 3 through Ref. 11, for example.

When there is no touch on the panel, the touch panel Z direction resistance R_Z is almost infinite. When a touch is put on the panel, it becomes conductive, with resistance from several hundred ohms to 1 kΩ, inversely proportional to the pressure on the touch panel (denoted by P). That is: the pressure on touch panel, P, can be expressed as a function of R_Z and it is given by Equation 9:

$$P = \alpha - \beta \times R_Z \tag{9}$$

Where α and β are positive real values and can be obtained by experience.

Note that the resolution of the touch pressure is very poor for the most touch panels currently on the market. Thus, the P from Equation 9 provides only the trend of the pressure on the touch panel.

The logic for the host to use the pressure P and validate a set of touch screen data from the TSC is shown in Example 5.

Example 5.

```
//*****************************************************************************//
unsigned int P;                  //  the touch pressure
unsigned int Threshold           //  the touch pressure threshold

// Sample X, Y, Z1 and Z2 touch data
// Calculate Rz using Eq.8.
// Calculate P using Eq 9.

if (P > Threshold)
{
//   the set of touch coordinates (X, Y) is valid
}
//******************************************************************************//
```

This method is very effective for reducing noise if the touch system is configured to ignore any light or non-firm touch. Many of TI's TSC EVM systems, such as that in the TSC2003EVM-PDK or the TSC2005EVM-PDK, use this approach to help reduce noise in TSC evaluation applications.

6 Conclusions

To reduce analog input noise in a resistive touch screen system, both hardware and software approaches are recommended. PCB layout is always the first line of defense against noise; the added noise-decoupling capacitor in the TSC input pin may also help when all other methods had been exhausted. Average/filter touch data from multiple samples are an efficient technique for reducing noise and refining TSC data. Selecting a TSC device with built-in filtering is one of the important considerations at system design. When using the host processor to perform average/filter functions, users may also apply several simple algorithms to reduce noise. The touch pressure threshold can also be used to eliminate unreliable data.

7 References

Unless otherwise noted, copies of the referenced data sheets and application reports can be downloaded from the Texas Instruments web site at www.ti.com.

1. Baker, B. and Fang, W. (2007.). *Powering resistive touch screens efficiently*. May 2007, Planet Analog.
2. Fang, W. (2007.).*Operation Schemes of Touch Screen Controllers*. Texas Instruments application note SLAA359.
3. ADS7843: Touch Screen Controller. Texas Instruments product data sheet SBAS090.
4. ADS7845: Touch Screen Controller. Texas Instruments product data sheet SBAS104.
5. ADS7846: Touch Screen Controller. Texas Instruments product data sheet SBAS125.
6. TSC2003: I^2C™ Touch Screen Controller. Texas Instruments product data sheet SBAS162.
7. TSC2046: Low Voltage IO Touch Screen Controller. Texas Instruments product data sheet SBAS265.
8. TSC2004: 1.6V to 3.6V, 12-Bit, Nanopower, 4-Wire Touch Screen Controller with I^2C™ Interface. Texas Instruments product data sheet SBAS408.
9. TSC2005: 1.6V to 3.6V, 12-Bit, Nanopower, 4-Wire Touch Screen Controller with SPI™ Interface. Texas Instruments product data sheet SBAS379.
10. TSC2006: Nanopower, 4-Wire Touch Screen Controller with SPI™ Interface. Texas Instruments product data sheet SBAS415.
11. TSC2007: 1.2V to 3.6V, 12-Bit, Nanopower, 4-Wire Micro Touch Screen Controller with I^2C™ Interface. Texas Instruments product data sheet SBAS405.
12. Fang, W. (2007.). *Set up and use TSC timings*. Texas Instruments application note SLAA362.

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 TI 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. Information of third parties may be subject to additional restrictions.

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.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

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
RFID	www.ti-rfid.com	Telephony	www.ti.com/telephony
Low Power Wireless	www.ti.com/lpw	Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2007, Texas Instruments Incorporated