LDC2112 and LDC2114 Internal Algorithm Functionality

Chris Oberhauser

ABSTRACT

This Application Note details the operation and usage of the LDC211x internal algorithms. It covers what conditions are addressed by a given algorithm and the parameters that control each algorithm. With the information provided here, the system designer can obtain the optimum performance out of a given design.

Contents

1 Introduction ................................................................................................................... 2
2 Scan Rate and Sampling Interval ............................................................................. 2
3 Data Polarity and Timeout ....................................................................................... 5
4 Internal Algorithms Overview ............................................................................... 6
5 Baseline Tracking ..................................................................................................... 8
6 Gain, Hysteresis, and Threshold ............................................................................. 14
7 Multi-Button Algorithms ........................................................................................ 15
8 Summary .................................................................................................................... 21

List of Figures

1 Sample Interval and Scan Rate.............................................................................. 2
2 Sampling Flow Chart ................................................................................................. 4
3 DPOlx functionality with Inductive and Capacitive Sensing ................................ 5
4 Algorithm Sequence ................................................................................................. 7
5 Baseline increment produces a decaying output code ............................................. 8
6 Effect of different values of Baseline increment ..................................................... 9
7 Baseline Tracking Reset during button press .......................................................... 10
8 Effect of Baseline Increment on Minimum Actuation time ...................................... 11
9 BTpause Functionality .............................................................................................. 12
10 Effect of Fast Tracing on Output Code .................................................................... 13
11 LDC2112/4 Gain Factor vs. Programmed Register Field Setting ......................... 14
12 Compare Thresholds based on HYST setting ......................................................... 15
13 Max-Win functionality .............................................................................................. 16
14 Anti-Common Functionality .................................................................................. 17
15 Example Twist Force Applied to Device ................................................................. 18
16 Anti-Twist Functionality ........................................................................................ 19
17 Deformation Can Cause Positive or Negative Offsets .......................................... 20
18 Anti-Deform Functionality ..................................................................................... 21

List of Tables

1 LDC211x Sample Control .......................................................................................... 2
2 LDC211x Internal Algorithms ................................................................................. 6
3 Baseline Tracking Clear and Reset ........................................................................ 10
1 Introduction

The Texas Instruments LDC2112 and LDC2114 devices can monitor changes in a sensor resonant frequency to determine whether a stimuli corresponds to a user interaction such as a button press. Refer to the LDC2112/4 Design Guide for details on how to construct a mechanical system suitable for inductive sensing button applications, sensor design guidance, and an overview of device configuration. The LDC211x uses several internal algorithms to compensate for environmental shifts, determine whether the sensor signal change corresponds to a button press, and also to correct for mechanical cross-talk between multiple buttons. This document details the operation and configuration of the algorithms.

2 Scan Rate and Sampling Interval

Once the LDC2112/4 powers on, it continuously samples the enabled channels and returns to as shown in Figure 1. The active channels, sample time interval, and button scan rate are all configurable parameters. Refer to Table 1 for a listing of the device parameters that control the Scan Rate and Sample Interval. A higher Scan Rate produces a more responsive interface, but generally at the cost of higher average supply current consumption. Note that the scan rate can vary by up to ±30% from the nominal value across devices.

![Figure 1. Sample Interval and Scan Rate](image)

**Table 1. LDC211x Sample Control**

<table>
<thead>
<tr>
<th>Device Parameter</th>
<th>Configuration Effect</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>SENCYCx</td>
<td>Button Sample Interval for channel x</td>
<td>Each channel has dedicated setting. Sensor frequency is used to determine proper setting.</td>
</tr>
<tr>
<td>CNTSCx</td>
<td>Button Sample Interval for channel x</td>
<td>Each channel has dedicated setting. Sensor frequency, LCDIV, and SENCYCx is used to determine proper setting</td>
</tr>
<tr>
<td>LCDIV</td>
<td>Button Sample interval for all channels</td>
<td>Common setting for all channels</td>
</tr>
<tr>
<td>LPWRB pin</td>
<td>Button Scan Interval</td>
<td>When set High, device samples based on NPSR value. When set Low, device sample based on LPSR value.</td>
</tr>
<tr>
<td>NPSR</td>
<td>Button Scan Interval</td>
<td>When LPWRB pin is High, this field sets the device scan rate from 10 sps to 80 sps.</td>
</tr>
<tr>
<td>LPSR</td>
<td>Button Scan Interval</td>
<td>When LPWRB pin is Low, this field sets the device scan rate from 0.625 sps to 5 sps.</td>
</tr>
</tbody>
</table>
The Button Sample Interval is a function of the SENCYCx, LCDIV, and the Sensor Oscillation Frequency. The sample time, $t_{\text{SAMPLE}}$, is set by:

$$t_{\text{SAMPLE}} = 128 \times (\text{SENCYCx}+1) \times 2^{\text{LCDIV}} \div f_{\text{SENSORx}}$$

The recommended sample interval is 1 ms, as this provides a good balance on noise vs. supply current. Shorter sample intervals may have reduced SNR. Variations in sensor frequency greater than 2.5% may need to use different settings for SENCYCx. If the sensor frequency increases significantly and the SENCYCx & LCDIV settings are not changed, then a reduction in sensitivity can occur. Alternatively, if the sensor frequency decreases by ~30% from the configured frequency, it is possible for internal counters in the device to over-range, resulting in improper operation.

2.1 Low Power Mode and Normal Power Mode

The LDC2114 operates in one of two modes – Low Power Mode and Normal Power Mode. In Low Power Mode, the maximum supported scan rate is from 0.625 to 5 sps. In Normal Power Mode, the scan rate is 10 to 80 sps. The primary difference between the two modes is the Baseline Increment, which is discussed below. The buttons enabled in Low Power Mode must be a subset of the buttons enabled in Normal Power Mode.

2.2 Button Sequencing and Error Handling

The LDC211x starts sampling channel 0 and checks to see if it is enabled. If channel is enabled, it starts the sensor oscillation and checks whether the sensor is oscillating. If so, it measures the sensor frequency, otherwise the device flags an error. It then increments to the next channel, and repeats the process. Figure 2 represents this sequence as a flowchart.

If the sensor cannot oscillate in a stable manner, the device flags the error in STATUS:LC_WD and continues onto the next channel. Potential sources of a LC_WD error include the sensor frequency exceeds the sensor maximum frequency of 30 MHz, or sensor frequency is below the minimum specified sensor frequency of 1 MHz, or the sensor $R_p$ is less than the minimum supported $R_p$.

The LDC211x device is intended to maintain operation under a wide range of conditions, and it may not generate an error condition even if the sensor is not within the specified operating region. Refer to Figure 4 for details on the Execute Algorithms operation shown in Figure 2.
Start

Bias COM and enable measurement system

Set Channel = 0

Is channel enabled in current mode?

Initiate Sensor Drive

Is Sensor oscillating?

Measure Sensor Frequency

Flag Error and disable sensor drive

Increment Channel

Is channel =3?

Execute Algorithms

Turn off COM and disable measurement system

End

Figure 2. Sampling Flow Chart
3 Data Polarity and Timeout

The LDC211x can detect positive or negative frequency shifts as button events. This is controlled per channel by the DPOLx setting, which adjusts the polarity of data processing. Figure 3 shows how this can be configured to support either inductive or capacitive sensing by the same device, or even a combination where some buttons operate as inductive sensors and the others operate as capacitive sensors. For a capacitive sensor, setting DPOLx to 0 inverts the sign of the LDC211x internal data processing. This results in a signal which operates in the same sign as an inductive sensor, and so all other internal algorithms operate consistently.

3.1 Button Timeout

The maximum supported continuous button actuation is ~50 seconds. If the button is asserted for a longer time interval, the device internal state machine automatically resets operation and the baseline tracking value automatically resets, which deasserts the button. When a timeout occurs, the LDC211x will flag the event by setting STATUS:TIMEOUT to 1.
4 Internal Algorithms Overview

The LDC211x device takes the output measurements for each channel and applies a sequence of several algorithms to the conversion results. Some of the algorithms are optional, and some can be configured per-channel. Table 2 below lists the internal algorithms and the configuration options applicable for each algorithm.

Table 2. LDC211x Internal Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Always Applied or Optional?</th>
<th>Configuration</th>
<th>Per Channel</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Polarity</td>
<td>Always</td>
<td>DPOLx</td>
<td>Y</td>
<td>Select Inductive or Capacitive sensing operation</td>
</tr>
<tr>
<td>Baseline Tracking</td>
<td>Always</td>
<td>NPBI/LPBI</td>
<td>N</td>
<td>Compensates for environmental shifts in inductance and capacitance.</td>
</tr>
<tr>
<td>Fast Tracking</td>
<td>Optional</td>
<td>FTF</td>
<td>Y</td>
<td>Provides faster recovery for negative swings in output code values.</td>
</tr>
<tr>
<td>Gain</td>
<td>Always</td>
<td>GAINx</td>
<td>Y</td>
<td>Adjusts sensitivity of channel. Scales between 1x and 232x in 64 settings; average step delta is 9%.</td>
</tr>
<tr>
<td>Threshold Compare</td>
<td>Always</td>
<td>N/A</td>
<td>N</td>
<td>Centered at 128. Effectively adjusted by use of GAINx.</td>
</tr>
<tr>
<td>Hysteresis</td>
<td>Always</td>
<td>HYST</td>
<td>N</td>
<td>Sets button actuation/deactivation thresholds. Centered at fixed Threshold Compare level of 128. Adjustable from 0 to 60 in steps of 4.</td>
</tr>
<tr>
<td>Baseline Tracking Pause</td>
<td>Optional</td>
<td>BTPAUSEEx</td>
<td>Y</td>
<td>Disables Baseline Tracking when OUTx value exceeds THRESHOLD+HYSTERSIS.</td>
</tr>
<tr>
<td>Max Win</td>
<td>Optional</td>
<td>MAXWINx</td>
<td>Y</td>
<td>Compares OUTx across selected channels and deasserts OUTx signal for all but the highest value channel.</td>
</tr>
<tr>
<td>Anti-Common</td>
<td>Optional</td>
<td>ANTICOMMONx</td>
<td>Y</td>
<td>Suppress common-mode signal present on multiple channels which are due to mechanical cross-talk.</td>
</tr>
<tr>
<td>Anti-Deform</td>
<td>Optional</td>
<td>ANTIDFORMx</td>
<td>Y</td>
<td>Compensate the baseline tracking for non-ideal recovery from mechanical stresses.</td>
</tr>
<tr>
<td>Anti-Twist</td>
<td>Optional</td>
<td>ANTITWIST</td>
<td>N</td>
<td>Suppress inverse signals present on multiple channels which are due to mechanical cross-talk.</td>
</tr>
</tbody>
</table>
The various algorithms are applied in the sequence shown in Figure 4. For algorithms which are optional, if the algorithm is not enabled, then the channel data is not modified by block.

Figure 4. Algorithm Sequence
5 Baseline Tracking

5.1 Baseline Increment

Button detection applications can be effectively implemented by looking at a high-pass filtered version of the input signal, due to the typical stimuli. Refer to Figure 5. In the LDC211x devices, this high-pass filtering is implemented using a subtracted baseline tracking which outputs the shift from the nominal code. This baseline tracking is designed to ignore slow changes in the output code, as these are generally due to environmental shifts which occur over the span of seconds, while a user interaction typically occurs on the order of 50 ms.

The LDC211x uses a linear offset approach, which always drives the output code towards 0.

![Figure 5. Baseline Increment produces a decaying output code](image)

Each channel of the LDC211x has its own Baseline value, but the baseline increment is the same for all channels. The internal baseline setting of the LDC211x is updated for each sample, using the following pseudo-code:

```plaintext
If raw_data[channel] > baseline[channel]
    then
        baseline[channel] = baseline[channel] + base_increment

iIf raw_data[channel] < baseline[channel]
    then
        baseline[channel] = baseline[channel] - base_increment
```

The net code is then calculated from the raw code with:

```plaintext
net[channel] = raw_data[channel] - baseline[channel]
```

Where the net code is the output code of the device. Note that this net value will be scaled by the Gain_Factor and modified by other algorithms before storage in the output register. The baseline increment value can be adjusted across a range of 1 to 128, in 8 settings. A higher setting will bring the output code to 0 in less samples, as displayed in Figure 6.
Figure 6. Effect of different values of Baseline Increment

The baseline value is reset whenever the device changes modes (e.g. from Config Mode to Low Power Mode, or from Low Power Mode to normal power mode). Note that baseline tracking is applied for every sample. Faster sample rates have an effectively higher baseline tracking rate per unit time. The Baseline Increment for Normal Power Mode is:

Baseline Increment per Sample = \( \text{Gain\_Factor}_x \times \frac{1.827}{2^{(7-NPBI)}} \)

Where:
- \( \text{Gain\_Factor}_x \) is the linear gain value selected for the channel based on the GAINx field setting, and
- \( NPBI \) is the value programmed into the NPBI field

To compensate for the slower sampling interval used in low power mode, the Low Power mode Baseline Increment is effectively 8x larger for the same setting:

Baseline Increment per Sample (Low Power Mode) = \( \text{Gain\_Factor}_x \times \frac{1.827}{2^{(6-LPBI)}} \)

Where:
- \( \text{Gain\_Factor}_x \) is the linear gain value selected for the channel based on the GAINx field setting, and
- \( LPBI \) is the value programmed into the LPBI field

The effective Baseline Increment in either mode is not restricted to an integer value, as the LDC2114 has higher internal resolution than is represented by the output code.
5.2 Baseline Tracking Reset

The baseline tracking is reset under several conditions. When a baseline tracking is reset, the baseline tracking value is set to the current measured raw_data. If a button is pressed when a baseline tracking reset occurs, the button press will be deasserted. The OUTx pin will immediately be deasserted. A baseline tracking reset can result in reduced sensitivity to subsequent button presses until the baseline tracking returns to 0.

In Figure 7, a baseline reset occurs at t=0.6 s. The OUTx button then immediately deasserts. The second button press occurring at t=1.5 s is not detected due to the offset in the baseline tracking. A stronger button press would be detected at t=1.5 s. The third button press at t=2.25 s is detected, as the baseline tracking has returned close enough to 0.

![Figure 7. Baseline Tracking Reset during button press](image)

Some operations reset the baseline tracking for a subset of channels while other operations reset all channels. Another type of reset is a clear – when the baseline tracking is cleared, a button press will take 2 scan intervals to detect a button press event.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Channels Affected</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPWRB pin toggle</td>
<td>All channels</td>
<td>Clears baseline. If a button remains pressed during toggling of LPWRB, the button press will be deasserted.</td>
</tr>
<tr>
<td>Soft Reset</td>
<td>All channels</td>
<td>Clears baseline</td>
</tr>
<tr>
<td>Exit from Config Mode</td>
<td>All channels</td>
<td>Clears baseline</td>
</tr>
<tr>
<td>Continuous Button press &gt; 50 sec</td>
<td>All channels</td>
<td>Reset baseline</td>
</tr>
<tr>
<td>Button Press detected in Max Win group</td>
<td>Unasserted channels in Max-Win group.</td>
<td>Reset baseline</td>
</tr>
<tr>
<td>Button Press detected in Anti-Deform</td>
<td>Unasserted channels in Anti-Deform group.</td>
<td>Reset baseline</td>
</tr>
<tr>
<td>Anti-Twist threshold is exceeded</td>
<td>All channels</td>
<td>Reset baseline</td>
</tr>
</tbody>
</table>
5.3 **Button Actuation Time**

The minimum button actuation time is the hysteresis value divided by the baseline increment. Figure 8 shows the output codes and OUTx assertion for the same stimuli with different baseline increment settings. Larger values of baseline increment can effectively “turn-off” a button before the actual stimuli has been removed.

![Figure 8. Effect of Baseline Increment on Minimum Actuation time](image)
5.4 **BTPAUSE**

BTPAUSE is an optional adjustment to the baseline tracking, which inhibits the baseline tracking operation for the duration of a button press. This functionality applies to the internal data used by the LDC211x and not just the OUTx pin operation. This can keep a button actuated for a longer time, and also be useful for some applications which post-process data. BTPAUSE can be enabled on all channels or any subset of channels.

![Figure 9. BTPAUSE Functionality](image-url)
5.5 Fast Tracking Factor

The negative swings which occur after a button actuation can reduce sensitivity for multiple sequential button presses. While BTPAUSE is often a more effective algorithm for applications in which this is an issue, Fast Tracking Factor provides an increased Baseline Increment when OUTx is negative. Figure 10 compares the operation with and without Fast Tracking Enabled for an example stimuli. Each channel can be set with an independent setting for Fast Tracking – FTFx field can be set between 0 and 3.

![Figure 10. Effect of Fast Tracing on Output Code](image)

The following pseudo-code shows the effective change in Baseline Tracking when Fast Tracking Factor is enabled:

```python
If raw_data[channel] > baseline[channel]
    then
        baseline[channel] = baseline[channel] + base_increment

If raw_data[channel] < baseline[channel] = FTH) then
    baseline[channel] = baseline[channel] - FTF * base_increment

Else If raw_data[channel] < baseline[channel] then
    baseline[channel] = baseline[channel] - base_increment
```

The value of FTF is set by:

```
FTF = 2^{FTH*Field Value}
```

Where FTH is a correction to the FTF based on the device gain.

Fast Tracking is a multiplier onto the Baseline tracking setting. It is applied in both Normal Power mode and Low Power mode with the same scaling factor.
Gain, Hysteresis, and Threshold

Each channel has a dedicated gain setting (GAINx). The Gain setting can be adjusted from an effective 1x to 232x, in 64 steps. Throughout this document, the value programmed into the register is referred to as GAINx, and the effective gain value that is applied by the algorithm is referred to as the Gain Factorx. The steps are logarithmically spaced, with each increment in the GAINx field increasing the Gain Factor by ~9%. Using a lower Gain Factor reduces the sensitivity to environmental shifts. The gain is applied to net code, which is the raw measured value with the baseline tracking subtracted.

The effective Net Code shift to a fast stimuli is:

\[ \text{NET\_CODE}_{\text{ex}} = 0.013 \times \text{Gain\_Factor}_{\text{x}} \times \Delta f_{\text{SENSOR}} \]

Where \( \Delta f_{\text{SENSOR}} \) is in PPM
6.1 Threshold and Hysteresis

After applying Gain Factor $x$, each channel $Net_x$ is compared to a fixed threshold of $128 + Hysteresis$. The Hysteresis value is set by the HYST register field. The Hysteresis value can be set from 0 to 60, in increments of 4. The HYST setting is common to all channels. A larger HYST setting improves the noise immunity and also increases the minimum button actuation time (when BTPAUSEx is not enabled). The GAINx and HYST settings effectively produce a variable Upper and Lower threshold, as seen in Figure 12.

![Figure 12. Compare Thresholds based on HYST setting](image)

The hysteresis and threshold operation is consistent with standard practice, in that:

- If OUTx is asserted, OUTx deasserts if NETx < Lower Threshold
- If OUTx is deasserted, OUTx asserts if NETx > Upper Threshold

7 Multi-Button Algorithms

The LDC211x includes several algorithms which operate on multiple channels. These algorithms can be used to suppress false button actuation which is caused by mechanical cross-talk between multiple channels. Each of these algorithms can be applied to any subset of buttons, although a set of a single button will not have any effect. In general, it is not recommended to apply multiple algorithms to the same channel, although using one algorithm on one pair of channels and a different algorithm to the other pair of channels with the LDC2114 is acceptable.

7.1 Max Win

Max-Win is a useful algorithm which can be used when significant positively-correlated mechanical cross-talk is present in the system. This algorithm can be used on any subset of buttons, but is only effective if there are 2 or more enabled channels. This algorithm behaves identically in both Low Power Mode and Normal Power Mode. Note that this algorithm only modifies the assertion of the OUTx pins and the OUTx field; although the baseline tracking reset it performs will affect subsequent samples.
To use Max-Win, set the desired MAXWINx fields to 1. As this algorithm uses the post-gain value, adjusting the Gainx value affects the performance of this operation. The algorithm compares all enabled buttons with MAXWINx enabled that have net code exceeding the threshold. In the case of an equal net value across 2 or more channels, the lower indexed channel dominates (e.g. channel 0 will suppress channel 1). Any suppressed channels have their baseline tracking value updated to the current setting.

This algorithm can also be used to inhibit up to 3 channels with a reference sensor which has a higher Gain Factorx. By constructing the reference sensor in a manner which it is sensitive to extreme torque or twist events, false button actuations can be suppressed.
### 7.2 Anti-Common Mode

Anti-Common Mode corrects for unintended mechanical deflection which is common to multiple buttons. Figure 14 provides a comparison between having Anti-Common enabled vs. disabled for an example stimulus.

![Diagram showing conversion code, time, and output with and without Anti-Common](image)

**Figure 14. Anti-Common Functionality**

Buttons can be included in the Anti-Common group by setting the corresponding ANTICOMx field to 1. The default setting for the ANTICOMx field is 0, which disables Anti-Common for corresponding channel. The pseudo-code for this algorithm is equivalent to:

```plaintext
ACM_offset = (∑net[channel]) / num_acm_chan

For each (channel with enabled ACM)

net[channel] = net[channel] - ACM_offset

Next channel

Note that the Net output value readback from the LDC211x is changed by the use of this algorithm.
7.3  **Anti-Twist Factor**

Anti-Twist inhibits false button events which are caused by torques and twists on the unit. These forces can produce opposite-phase responses on buttons.

![Figure 15. Example Twist Force Applied to Device](image)

This functionality is enabled by setting the ANTITWIST register field to any value from 1 to 7; it is not enabled if ANTITWIST is set to 0, which is the default value. When ANTITWIST is enabled, all active buttons will be affected by the processing. The Anti-twist threshold is $4 \times \text{ANTITWIST}$, and so it can be set from 4 to 28. Anti-Twist uses the following pseudo-code:

```plaintext
If Out[0] or Out[1] or Out[2] or Out[3] is true then
    If at least for one channel: net[channel] > AntTwist and
    If at least for one channel: net[channel] < -AntTwist then
    For each active channel
        baseline[channel] = raw_data[channel]
    Next Channel
```
Twist applied to unit produces out-of-phase response on Button 0 & Button 1

Intentional Press of Button 1

Threshold+HYST

Anti-Twist Threshold

Channel0

Channel1

Channel0 Out

Channel1 Out

OUTPUT without Anti-Twist (high = Button Press Detected)

OUTPUT with Anti-Twist (high = Button Press Detected)

Figure 16. Anti-Twist Functionality

Basically, if any channel is found to exceed the Anti-Twist threshold in a positive direction simultaneous with another channel exceeding the Anti-Twist threshold in a negative direction, then any button press detection is disabled. This algorithm does not alter the Net Data in the output registers.
7.4 **Anti-Deform Factor**

Anti-Deform compensates for common-mode shifts on other buttons when a button is pressed. The common mode shifts are caused by mechanical offsets, and these offsets can either increase the likelihood of a false actuation or significantly increase the required mechanical actuation force.

![Diagram of anti-deform compensation](image)

**Figure 17. Deformation Can Cause Positive or Negative Offsets**

Anti-Deform resets the baselines for channels that don’t have a button press detection. This returns the sensitivity back to the system nominal level.
Channels can be included in the Anti-Deform group by setting the corresponding ANTIDFORMx field to 1. The default setting for the ANTIDFORMx field is 0, which disables Anti-Deform for corresponding channel.

The pseudo-code for this algorithm is equivalent to:

```plaintext
If channel is active and channel has ANTIDFORM enabled and OUT[channel] is true then
For each AntiDform channel
  baseline[channel] = raw_data[channel]
Next
```

8 Summary

The internal algorithms of the LDC211x devices can address a wide range of system variations. Appropriate use of the algorithms and details on the functionality provide clarity in which conditions to use which algorithm.
IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated (‘TI”) technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, “TI Resources”) are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI’s applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT. AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources 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.

TI RESOURCES ARE PROVIDED “AS IS” AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include, without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2018, Texas Instruments Incorporated