Application Report
SLAA421A–October 2009–Revised May 2018

Migrating From MSP430F12x(2) to MSP430F21x2 MCUs

MSP430 Applications

ABSTRACT

This application report facilitates the migration of application designs based on the MSP430F122, MSP430F123, MSP430F1222, or MSP430F1232 microcontrollers (MCUs) to the MSP430F21x2 MCU family. This document describes the main differences between the two MCU families and describes migration solutions for both software and hardware.

Contents

Trademarks

1 Comparison of MSP430x1xx and MSP430x2xx MCUs

The MSP430x2xx MCUs provide an upgrade path for the MSP430x1xx family, offering more performance, lower power, and more built-in features. These upgraded features enable an improved and more cost-optimized system design. Table 1 compares the two MCU families and gives reasons to consider migrating.

Table 1. Comparison of MSP430x1xx and MSP430x2xx MCUs

Feature or Function	MSP430x1xx	MSP430x2xx
Maximum CPU clock speed	8 MHz	16 MHz
Wake up from LPM3 or LPM4	<6 µs	<1 µs
Standby current consumption (LPM3)	<2 µA	<1 µA
Brownout reset (BOR)	Some MCUs	All MCUs
Minimum voltage for flash ISP	2.7 V	2.2 V
Integrated port pullup or pulldown resistors	–	On all ports
Internal oscillator (DCO)	Large voltage and temperature drift (±20%)	Very small voltage and temperature drift (±2%) Factory calibrated
Oscillator fault detection	High-frequency crystal	High-frequency and low-frequency crystals
Additional built-in low-power low-frequency oscillator	–	12-kHz VLO
Additional oscillator features	–	Minimum pulse clock filter for increased system robustness Configurable built-in crystal load capacitors
Additional watchdog timer features	–	Invalid address detection Fail-safe clock source
Bootloader (BSL)	Protected by 256-bit password	Hack proof
Flash memory configurations	Up to 60KB	Up to 120KB
RAM	Up to 10KB	Up to 8KB
Operating temperature (T_A)	–40°C to 85°C	–40°C to 85°C (up to 105°C or 125°C on selected MCUs)

While the MSP430F21x2 MCUs can be considered as a direct pin-to-pin compatible drop-in to existing MSP430F12x(2) designs, some important details require attention. This application report helps identify potential issues. After migration, the MSP430F2xx family enhancements in Table 1 improve the application. The improvements might enable further cost savings or other system-level optimizations. This document describes transitioning existing designs and leaves it to the engineer to make use of additional MSP430F2xx features during migration as applicable for a given system (for example, the use of internal pullup and pulldown resistors or the use of peripheral modules that are not available on the original MCU).

2 MSP430F12x(2) to MSP430F21x2 Migration – Hardware Considerations

This section provides information on differences between the MSP430F12x(2) and MSP430F21x2 MCUs that should be considered during migration. Fortunately, the hardware migration process is straightforward with only a few items to consider.

2.1 Device Package and Pinout

The good news is that a 28-pin TSSOP and the 32-pin QFN MSP430F21x2 MCUs directly drop into existing MSP430F12x(2)-based footprints. Both TSSOP and QFN package variants and PCB footprints are identical. However there is no 28-pin DW package equivalent for the MSP430F12x(2) MCUs in the MSP430F21x2 MCU family. If the original design uses the 28-pin DW package, the PCB must be migrated to use either the 28-pin TSSOP or the 32-pin QFN package variant of the MSP430F21x2.

All MSP430F21x2 pins can be used for the same purpose as the pins on their MSP430F12x(2) counterparts (which includes all analog and digital modules, as well as power supply pins), enabling a transition to MSP430F21x2 MCUs without changes to the application PCB. Details regarding packaging and pinouts can be found in the device-specific data sheets. [3][4][5]

The MSP430F21x2 MCUs support a 2-wire JTAG communication interface (also referred to as Spy-Bi-Wire) in contrast to the MSP430F12x(2) MCUs, which have a 4-wire JTAG communication mode-only interface. The MSP430F21x2 family also supports 4-wire JTAG communication, but the connection of the $\overline{\text{RST}}$ pin is not optional. If a custom board relies on the JTAG interface for in-system programming purposes, this connection must be added to the circuit. For more information about the necessary connections, see the MSP430 Hardware Tools User's Guide.

2.2 Current Consumption

When migrating to an MSP430F21x2, the difference in current consumption of the MCUs should be considered. For example, in LFXT1 standby mode (LPM3 using a 32-kHz watch crystal), the standby current consumption of an MSP430F21x2 is in the 1-µA range (typical data sheet value at 3 V, 25°C), which is much lower than the current consumption of an MSP430F12x(2) MCU, which is in the 1.6-µA range under the same conditions. This reduced consumption is a benefit for applications that operate in standby mode most of the time.

Due to architectural differences, the active mode current consumption of the MSP430F21x2 MCUs is approximately 15% higher (typical) than MSP430F12x(2) MCUs when operating at the same frequency, temperature, and voltage conditions. See the device-specific data sheets for the exact specifications.

When using LFXT1 in high-frequency mode, the current consumption by the oscillator of an MSP430F21x2 MCU is slightly higher than an MSP430F12x(2) MCU due to changes in the oscillator design to support higher frequencies.

When taking advantage of the increased maximum operating frequency of the MSP430F21x2, additional current must be supplied by the system power supply, because active mode current consumption scales linearly with operating frequency.

2.3 Operating Frequency and Supply Voltage

The maximum frequency at which the MSP430™ CPU can operate depends on the supply voltage. This specification can be found in the recommended operating conditions of each data sheet. In general, this specification differs for the MSP430F12x(2) and MSP430F21x2 MCUs. However, an MSP430F21x2 MCU can always operate under the same operating conditions in terms of supply voltage and CPU clock frequency (MCLK) as an MSP430F12x(2) MCU. When using the increased maximum clock frequency of an MSP430F21x2 MCU, review the recommended operating conditions in the MSP430F21x2 Mixed-Signal Microcontrollers data sheet.

It is of extreme importance that this relationship is also observed during power-ramp scenarios. Violating the maximum frequency versus voltage dependency can result in unpredictable code execution. Possible solutions include the addition of system-specific delays during MCU startup and measuring the supply voltage using the built-in ADC10 module before increasing the system frequency. The use of an external SVS should be considered for the most robust system.

2.4 Device Errata

When migrating an existing application to the MSP430F21x2, see the most current MCU errata sheets to ensure the application is not affected by a known issue. The errata sheets typically include workarounds with the bug descriptions. For all MSP430 products, the MCU errata sheets can be found in the product folders of each product on the MSP430 web page.

3 MSP430F12x(2) to MSP430F21x2 Migration – Firmware Considerations

This section describes important considerations when migrating existing software routines or an entire application to an MSP430F21x2 MCU. Even though the MSP430F12x(2) and MSP430F21x2 MCUs are code compatible and share many of the same peripherals, migration is not as simple as programming the MSP430F12x(2) binary image into an MSP430F21x2 MCU. An application must be rebuilt at a source-code level (including all referenced code libraries), using the appropriate MSP430F21x2 MCU support files, such as the header file (msp430x21x2.h) and the linker command file (for example, lnk_msp430f2132.cmd for TI Code Composer Studio™ IDE or lnk430F2132.xcl for IAR Embedded Workbench® IDE). This is the first step toward a successful migration to an MSP430F21x2 MCU. The following sections describe other aspects to consider.

3.1 Memory Considerations

3.1.1 Device Memory Map

The memory maps of the MSP430F12x(2) and MSP430F21x2 are almost identical. This applies to the location and size of RAM as well as flash memory, which might allow an application to keep the same linker command file during migration. However, there are two exceptions in the memory map and, therefore, TI strongly recommends to rebuild the application with the correct linker file to accommodate for the difference in the memory map. The build process makes use of the memory map information stored in the IDE linker command file and automatically accommodates these changes. The linker command files are installed with the IDE and have the file name extension CMD (for TI Code Composer Studio IDE) and XCL (for IAR Embedded Workbench IDE).

The interrupt vector table of MSP430F21x2 MCUs spans 32 memory word locations, and the table in MSP430F12x(2) MCUs spans 16 memory word locations. The word memory location 0xFFDE on MSP430F21x2 MCUs is reserved for special BSL purposes (BSLSKEY). See Section 3.5 for more details about the interrupt vector table.

Both MSP430F2122 and MSP430F2132 MCUs have an increased RAM size of 512 bytes compared to their MSP430F12x(2) family counterparts. The application should be rebuilt to take advantage of this increased memory size. In addition, the MSP430F2112 MCU has 256 bytes of RAM and 2KB of flash memory, which is a memory configuration previously unavailable. This MCU can be considered as an alternative migration option for applications that do not use the entire flash memory of the original device.

Further details regarding the MCU memory maps can be found in the device-specific data sheets.[3][4][5]

3.1.2 Information Flash Memory

Both MSP430F12x(2) and MSP430F21x2 MCUs have 256 bytes of information flash memory located in the memory range of 0x1000 to 0x10FF. While the total memory size is the same, the memory is organized differently. The MSP430F12x(2) MCU information memory consists of two flash segments (INFOA and INFOB) that are 128 bytes each, and the MSP430F21x2 has four segments (INFOA, INFOB, INFOC, and INFOD) that are 64 bytes each.

Applications that store data in the information memory must be modified for the different segment sizes. Each information flash memory segment must be erased individually, resulting in four write accesses on an MSP430F21x2 MCU instead of two on an MSP430F12x(2) MCU. Also, the MSP430F21x2 INFOA segment is protected by a lock feature and requires special treatment to be erased or written. In most cases, TI recommends not to erase INFOA or store any user data in it. INFOA on an MSP430F21x2 MCU contains factory-provided device-specific calibration data organized in a tag-length-value (TLV) structure, which can help to generate specific frequencies using the DCO or to calibrate the ADC10. Many applications can benefit from those constants.

See the MSP430x2xx Family User's Guide for more details on the organization of the 2xx information flash memory, the TLV structure, and the INFOA lock feature.

3.2 Serial Communication – USART and USCI

One of the major differences between MSP430F12x(2) and MSP430F21x2 MCUs is the serial communication module. On the MSP430F21x2 MCUs, the USCI module is implemented. The USCI module is the next-generation MSP430 communication module, offering more features and functionality that the USART module. USART and USCI modules are not software compatible. Therefore, MSP430F12x(2) software for the USART module must be changed to use the MSP430F21x2 USCI module.

The MSP430F21x2 supports one USCI module with two communication channels that can operate simultaneously. For example, the MSP430F21x2 MCU can simultaneously service two SPI communication channels or one I^2C channel and one UART channel. I^2C operation was not available on MSP430F12x(2) MCUs.

It is not in the scope of this application report to discuss all possible aspects regarding the migration of application code to use the USCI interface; however, the following sections describe the major differences between the MCUs (and the modules). TI strongly recommends that you review the module descriptions in the appropriate MCU family user's guide [1][2] and the USCI code examples provided in the product folders on the MSP430 web page as a starting point for any code that is newly created.

3.2.1 UART Mode

The operation of the MSP430F21x2 USCI in UART mode and that of the MSP430F12x(2) USART are almost identical. The major differences are:

- The MSP430F21x2 USCI uses a different baud rate generator. A new modulation scheme provides a two-stage modulator that can implement an oversampling baud rate generation scheme. During application migration, the baud rate register settings must be recalculated. However, the USCI module can generate the same target baud rate using the same clock source as the MSP430F12x(2) USART.
- Start edge detection and clock activation are different on the two MCUs.

  The MSP430F21x(2) MCUs features a simplified process in which the USCI module automatically activates the USCI module clock source on start edge detection and then provides an interrupt to wake the CPU after an entire character has been received.

  On the MSP430F12x(2) UART, an interrupt is generated on start edge detection, and the application must activate the clock source and manage the character reception.
- On the MSP430F21x2 USCI, interrupt flags are no longer cleared automatically upon entering the interrupt service routine.

3.2.2 SPI Mode

In SPI mode, the operations of the MSP430F21x2 USCI and the MSP430F12x(2) USART are almost identical. The major differences are:

- The MSP430F12x(2) USART supports one channel of SPI communication (USART0), and the MSP430F21x2 USCI supports two channels (USCI_A0 and USCI_B0).
- On the MSP430F21x2 USCI, interrupt flags are no longer cleared automatically when entering the interrupt service routine.
- The MSP430F21x2 USCI defaults to an LSB-first SPI bit order. The bit order can be configured with the UCMSB bit in the UCA0CTL0 and UCB0CTL0 control registers. This is different from the USART module, in which the bit order is MSB first and cannot be configured.
- The maximum MSP430F21x2 USCI bit clock frequency in SPI master mode is BRCLK, and on the MSP430F12x(2) USART module it is BRCLK/2.

3.3 Clock System

3.3.1 LFXT1 Oscillator

The oscillator block of the MSP430F21x2 MCUs supersedes the ones found on MSP430F12x(2) MCUs. The MSP430F21x2 oscillator can operate with the same low- and high-frequency oscillators and clock sources, but consumes less power in low-frequency (LF) mode while providing increased robustness through a higher oscillation allowance. In addition, built-in software-configurable crystal load capacitors are provided in LF mode. The power-on default for the effective load capacitance in LF mode is 6 pF, which is in line with the MSP430F12x(2) LF oscillator.

When migrating designs that use external crystals or clock sources:

- The ability of MSP430F21x2 MCUs to detect low-frequency oscillator failures and indicate them by setting the LFXT1OF flag results in another path for the global oscillator fault flag (OFIFG) to become set. This may prevent the CPU from being clocked by a crystal or an external clock source in certain scenarios.
- If the existing MSP430F12x(2) design uses an external 32-kHz crystal for low-power mode operation and periodic wakeup (LPM3), and crystal-accurate precision is not required, the MSP430F21x2 built-in VLO oscillator can be used instead, resulting in the elimination of the external crystal and a reduced LPM3 power consumption. The VLO frequency is approximately 12 kHz (data sheet typical value) but can be measured and calibrated. For more details, see Using the VLO Library.
- If an external digital clock source is used, the newly available direct digital clock input mode on the MSP430F21x2 MCUs should be used (by setting the LFXT1S1 and LFXT1S0 control bits).
- If the existing MSP430F12x(2) design uses a high-frequency crystal or resonator on LFXT1, the appropriate frequency range must be configured in the MSP430F21x2 clock system control register BCSCTL3. The default range setting is for use with 0.4-MHz to 1-MHz crystals or resonators. For more details, see the *Basic Clock Module+* chapter in the MSP430x2xx Family User's Guide.
- See MSP430 32-kHz Crystal Oscillators and verify that the hardware design recommendations in this application report are closely followed in your application. Make changes if necessary to ensure the best application performance.

3.3.2 Digitally Controlled Oscillator (DCO)

The MSP430F12x(2) and MSP430F21x2 MCUs have different DCO modules. The MSP430F21x2 DCO offers higher accuracy, an extended frequency range allowing operation of the MCU up to the maximum operating frequency, and factory-provided calibration constants to facilitate the design of systems that operate without external clock sources.

When migrating designs that use the DCO:

- The default DCO frequency of an MSP430F12x(2) MCU is in the 800-kHz range, but it is in the 1.2-MHz range for an MSP430F21x2 MCU. This needs to be considered for applications that run the MCU using the default DCO settings.
- On an MSP430F21x2 MCU, consider loading the factory-provided DCO calibration constants into the DCO control registers to achieve a deterministic and stable output frequency. The use of DCO calibration constants eliminates the need for software-FLL algorithms that are used in combination with an external clock source on MSP430F12x(2) MCUs.
- The MSP430F12x(2) has three bits to control the fundamental frequency range (RSELx in the BCSCTL1 register), and the MSP430F21x2 has four control bits. Care must be taken when porting algorithms such as a software FLL that modify these bits.
- If an MSP430F12x(2) application applies hard-coded DCOx, MODx, and RSELx values to the DCO control registers, this results in a different frequency range on an MSP430F21x2 MCU.
- When enabling the external resistor DCO bias feature (by setting DCOR in the BCSCTL2 register), the MSP430F21x2 DCO starts behaving the same as an MSP430F12x(2) DCO. In this mode, the same bit settings and external bias resistors result in the same frequency being generated. See the device-specific data sheets for further details.[3][4][5]

3.4 Bootloader (BSL)

MSP430F21x2 MCUs have a new BSL firmware with enhanced security features. Both MSP430F12x(2) and MSP430F21x2 MCU memory access is protected by a 256-bit password. However, only MSP430F21x2 MCUs erase the entire MCU flash memory contents (including the factory-provided calibration constants stored in the INFOA flash segment) on the first attempt to access the MCU with an incorrect password. This behavior is configurable and needs to be considered for applications that use the BSL interface to provide in-field software upgrade capability.

3.5 Interrupt Vectors

The interrupt vector arrangement of MSP430F12x(2) and MSP430F21x2 MCUs is identical for a given peripheral module and does not require special attention from a low-level perspective. In general, recompiling the MSP430F12x(2) application code using MSP430F21x2 MCU support files automatically populates the interrupt vector table according to the device-specific requirements. However depending on how the interrupt vectors are assigned, the symbolic names used to allocate the timer interrupts may need to be migrated (see Section 3.7 for more information).

The memory range that is reserved for interrupt vectors (interrupt vector table) differs between MSP430F12x(2) and MSP430F21x2 MCUs. For MSP430F12x(2) MCUs, this memory ranges from address 0xFFE0 to 0xFFFF (16 words), and for MSP430F21x2 MCUs, it ranges from 0xFFC0 to 0xFFFF (32 words). In addition to this, the word memory location 0xFFDE is reserved on MSP430F21x2 MCUs and used as the BSL security key (BSLSKEY) (see Section 3.4).

3.6 Reserved Bits

The MSP430F21x2 MCUs support several upgraded peripherals compared to the MSP430F12x(2) MCUs, including the BCS+ and the Comparator_A+. This added functionality is supported in part through the use of register bits that were reserved on the corresponding MSP430F1xx peripheral. Newer generation MSP430 MCUs such as the MSP430F21x2 use these bits to implement additional functionality. If left in the default state, the peripheral usually behaves the same as its MSP430F1xx counterpart. Take care to not unintentionally change some of these bits, which can be caused by migrated MSP430F12x(2) firmware.

For example, consider the following comparison of CACTL2 control register of Comparator_A and Comparator_A+:

Figure 1. Comparator_A CACTL2 Register, F1xx Devices

7	6	5	4	3	2	1	0
Reserved				P2CA1	P2CA0	CAF	CAOUT
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	r-(0)

Figure 2. Comparator_A+ CACTL2 Register, F2xx Devices

7	6	5	4	3	2	1	0
CASHORT	P2CA4	P2CA3	P2CA2	P2CA1	P2CA0	CAF	CAOUT
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	r-(0)

For example, if firmware that uses the MSP430F12x Comparator_A module sets bit 7, the bit is unused so the firmware runs. If this same code is executed on an MSP430F21x2 MCU, the Comparator_A+ inputs are shorted together internally.

3.7 Timer

MSP430F12x(2) MCUs have one 16-bit timer module with three capture/compare blocks (Timer_A3). MSP430F21x2 MCUs have two independent Timer_A modules, one 16-bit timer module with three capture compare blocks (Timer0_A3), and one 16-bit timer module with two capture/compare blocks (Timer1_A2). During migration, the Timer_A3 module of the MSP430F12x(2) can be substituted seamlessly by the Timer0_A3 module of the MSP430F21x2.

Because rebuilding the application software using the updated MS430F21x2-specific MCU support files is strongly recommended, it is necessary to migrate the definitions used to assign the interrupt vectors in C using the "#pragma vector =" compiler directive to the new names. The definitions TIMERA0_VECTOR and TIMERA1_VECTOR used on the MSP430F12x(2) must be changed to TIMER0_A0_VECTOR and TIMER0_A1_VECTOR on the MSP430F21x2.

Although the timer register names have also changed (for example, TACTL is renamed to TA0CTL), it is not necessary to change them in the source code, because the msp430x21x2.h MCU header file contains macros that map the MSP430F12x(2) register names to the MSP430F21x2 names.

An undocumented feature on the MSP430F12x(2) allows the Timer_A module to be used in capture mode to generate interrupts on input signal transitions with the timer in stop mode (MCx in TACTL is set to 00h). This feature is no longer available on MSP430F21x2 MCUs. To generate capture interrupts, the respective MSP430F21x2 timer must be running. In this specific use case, consider clocking the timer using a low frequency (for example, ACLK) to minimize power consumption.

3.8 Analog Comparator

On the comparator of MSP430F12x MCUs, disabling the digital port functionality for an I/O pin by setting the associated bit in the Comparator_A Port Disable (CAPD) register to prevent parasitic cross currents during analog measurements disables the digital CMOS input buffer. However, on MSP430F21x2 MCUs with Comparator_A+, setting a CAPDx bit disables both input and output buffers for that pin.

4 References

1. MSP430x1xx Family User’s Guide
2. MSP430x2xx Family User’s Guide
3. MSP430x12x Mixed-Signal Microcontrollers data sheet
4. MSP430F11x2, MSP430F12x2 Mixed-Signal Microcontrollers data sheet
5. MSP430F21x2 Mixed-Signal Microcontrollers data sheet
6. MSP430 32-kHz Crystal Oscillators
7. Using the VLO Library
8. MSP430 Hardware Tools User's Guide

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from October 23, 2009 to May 4, 2018	Page
• Editorial updates throughout document	1

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