MSP430F21x1 Device Erratasheet

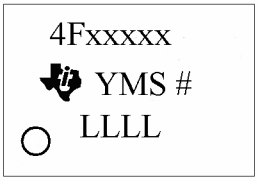
Current Version

Devices	Rev:	BCL12	CPU4	CPU11	CPU12	CPU13	FLASH16	FLASH17	FLASH18	FLASH19	FLASH20	FLASH24	TA12	TA16	XOSC5
MSP430F2101	I	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSP430F2111	I	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSP430F2121	I	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSP430F2131	I	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Note: See Appendix for prior revisions

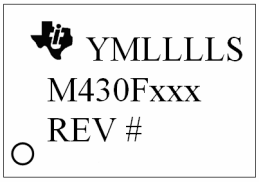
Package Markings

PW20: TSSOP(PW) 20-pin



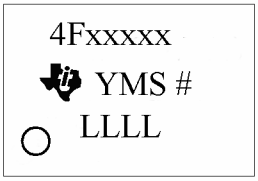
YM = Year and Month Date Code
LLLL = LOT Trace Code
S = Assembly Site Code
# = DIE Revision
o = PIN 1

DW20: SOP(DW) 20-pin

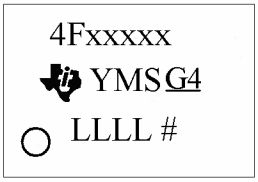


YM = Year and Month Date Code
LLLL = LOT Trace Code
S = Assembly Site Code
# = DIE Revision
o = PIN 1

DGV20: TVSOP(DGV) 20-pin



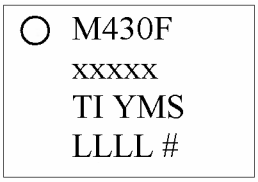
YM = Year and Month Date Code
LLLL = LOT Trace Code
S = Assembly Site Code
# = DIE Revision
o = PIN 1



YM = Year and Month Date Code
LLLL = LOT Trace Code
S = Assembly Site Code
# = DIE Revision
o = PIN 1

Package Markings (continued)

RGE24: QFN(RGE) 24-pin



YM = Year and Month Date Code
LLLL = LOT Trace Code
S = Assembly Site Code
# = DIE Revision
o = PIN 1

Detailed Bug Description

BCL12 BCL12 - Bug description

Module: BasicClock, Function: Switching RSEL can cause DCO dead time

After switching RSELx bits (located in register BCSCTL1) from a value of >13 to a value of <12, the resulting clock delivered by the DCO can stop before the new clock frequency is applied. This dead time is in the range of 20 µs.

Workaround:
Use an intermediate step when switching RSEL from >13 to <12. The intermediate RSEL value should be 13.

CURRENT RSEL	TARGET RSEL	RECOMMENDED TRANSITION SEQUENCE
15	14	Switch directly to target RSEL
14 or 15	13	Switch directly to target RSEL
14 or 15	0 to 12	Switch to 13 first, and then to target RSEL (two-step sequence)
0 to 13	0 to 12	Switch directly to target RSEL

CPU4 CPU4 - Bug description

Module: CPU, Function: PUSH #4, PUSH #8

The single operand instruction PUSH cannot use the internal constants (CG) 4 and 8. The other internal constants (0, 1, 2, –1) can be used. The number of clock cycles is different:
PUSH #CG uses address mode 00, requiring 3 cycles, 1 word instruction
PUSH #4/#8 uses address mode 11, requiring 5 cycles, 2 word instruction

Workaround implemented in assembler.
No fix planned

CPU11 CPU11 - Bug description

Module: CPU, Function: Invalid status register after program counter access

When addressing the program counter (PC) in register mode where the PC is the destination, the status register (SR) may be erroneous. The instructions BIS, BIC, and MOV do not affect SR contents. Only CPU flags are effected: DOES NOT apply to LPMx control bits.

Workaround:
None

Detailed Bug Description (continued)

CPU12 CPU12 - Bug description

Module: CPU, Function: CMP or BIT with PC destination

Any instruction immediately following a CMP(.B) or BIT(.B) instruction where the PC is the destination address using register mode is ignored or erroneously executed. When the following instruction is longer than one word, the second word is fetched by the CPU and decoded as the instruction, leading to unpredictable behavior. Affected source addressing modes are indexed and indirect addressing modes.
Example: cmp &200,PC
add #4,R8
The add command is not executed.

Workaround:
Insert a NOP instruction after the BIT or CMP instruction. The NOP is ignored, and program execution continues as expected.

CPU13 CPU13 - Bug description

Module: CPU, Function: Arithmetic operations and the SR

Performing arithmetic operations with the status register as the destination address does not update the status register as intended. The result in SR can be invalid, leading to erroneous low-power mode entry. Arithmetic operations are defined as all instructions that modify the SR flag bits (RRA, SUB, XOR, and ADD, for example).

Workaround:
None

FLASH16 FLASH16 - Bug description

Module: Flash, Function: Modifying INFOA addresses when LOCKA = 1 modifies main flash memory

When attempting to write to an address location or perform a segment erase of INFOA while the LOCKA bit is set, flash memory beginning at main memory location 0xF040 and extending for 64 bytes to address 0xF07F is modified erroneously. These 64 bytes are addressed and modified in place of the INFOA addresses when writes or erases are performed within the INFOA address space and LOCKA = 1.

Workaround:
Prior to modifying (writing or erasing) any address within the INFOA flash memory segment, properly clear the LOCKA control bit as described in the MSP430x2xx user's guide to unlock the segment. Once the modification is complete, setting the LOCKA bit is recommended.

Detailed Bug Description (continued)

FLASH17 FLASH17 - Bug description

Module: Flash, Function: Flash read when EEI = 1

When a segment erase in flash is initialized and the erase cycle is interrupted, any read from flash may be erroneous until after the erase cycle is completed.

Workaround:
When EEI = 1, initiate all flash segment erases using a byte access to an even address within the segment to be erased. For example:

(Assembly)

```
mov.b #0FFh, &1000h
```

(C)

```
*(char*)0x1000 = 0; //ERASE
```

FLASH18 FLASH18 - Bug description

Module: Flash, Function: INFO flash access when EEI = 1

When a segment erase to any flash location has been initiated and the erase cycle has been interrupted, it is not possible to read from the INFO flash memory locations until the erase cycle is completed.

Workaround:
None

FLASH19 FLASH19 - Bug description

Module: Flash, Function: EEI feature does not work for code execution from RAM

When the program is executed from RAM, the flash controller EEI feature does not work. The erase cycle is suspended, and the interrupt is serviced, but there is a problem while resuming with the erase cycle.
Addresses applied to flash are different than the actual values while resuming erase cycle after ISR execution.

Workaround:
None

FLASH20 FLASH20 - Bug description

Module: Flash, Function: Flash erase is not correctly executed when EEI is enabled and interrupts occur

Flash erase is not correctly executed when EEI is enabled, and an interrupt occurs in a certain time window. The BUSY signal from the flash controller and the "JMP $" feature, that keeps the program counter at the current position, disappear shortly after a flash erase cycle has been initialized.

Workaround:
None

Detailed Bug Description (continued)

FLASH24 FLASH24 - Bug description

Module: Flash, write or erase emergency exit can cause failures

When a flash write or erase is abruptly terminated, any further reliable reads by the flash controller are not ensured. The abrupt termination can be the result of one the following events:

1) The flash controller clock is configured to be SMCLK sourced by an external crystal. An oscillator fault occurs, thus stopping this clock abruptly.
Or
2) The Emergency Exit bit (EMEX in FCTL3) when set forces a write or an erase operation to be terminated before normal completion.
Or
3) The Enable Emergency Interrupt Exit bit (EEIEX in FCTL1) when set with GIE = 1 can lead to an interrupt causing an emergency exit during a flash operation.

Workaround:
1) Do not use SMCLK as the source for the flash controller clock if it is sourced from an external crystal.
2) After setting EMEX = 1, wait for a sufficient amount of time before flash is accessed again.
3) No Workaround. Do not use EEIEX bit.

TA12 TA12 - Bug description

Module: Timer_A, Function: Interrupt is lost (slow ACLK)

Timer_A counter is running with slow clock (external TACLK or ACLK) compared to MCLK. The compare mode is selected for the capture/compare channel and the CCRx register is incremented by 1 with the occurring compare interrupt (if TAR = CCRx). Due to the fast MCLK, the CCRx register increment (CCRx = CCRx + 1) happens before the Timer_A counter has incremented again. Therefore, the next compare interrupt should happen at once with the next Timer_A counter increment (if TAR = CCRx + 1). This interrupt is lost.

Workaround:
Switch capture/compare mode to capture mode before the CCRx register increment. Switch back to compare mode afterwards.

TA16 TA16 - Bug description

Module: Timer_A, Function: First increment of TAR erroneous when IDx > 00

The first increment of TAR after any timer clear event (POR/TACLR) happens immediately following the first positive edge of the selected clock source (INCLK, SMCLK, ACLK or TACLK). This is independent of the clock input divider settings (ID0, ID1). All following TAR increments are performed correctly with the selected IDx settings.

Workaround:
None

Detailed Bug Description (continued)

XOSC5 XOSC5 - Bug description

Module: LFXT1 OSC: LF crystal failures may not be properly detected by the oscillator fault circuitry

The oscillator fault error detection of the LFXT1 oscillator in low frequency mode (XTS = 0) may not work reliably causing a failing crystal to go undetected by the CPU, i.e., OFIFG is not set.

Workaround:
None

Appendix: Prior Versions

Devices	Rev:	BCL6	BCL8	BCL9	BCL10	BCL11	BCL12	BCL13	BSL5	CPU4	CPU5	CPU6	CPU11	CPU12	CPU13	CPU14	FLASH16	FLASH17	FLASH18	FLASH19	FLASH20	FLASH22	FLASH24	JTAG15	PORT8	PORT10	TA12	TA16	XOSC5
MSP430F2101	I						✓			✓			✓	✓	✓		✓	✓	✓	✓	✓		✓				✓	✓	✓
	H						✓	✓		✓			✓	✓	✓		✓	✓	✓	✓	✓		✓				✓	✓	✓
	G			✓	✓	✓	✓	✓		✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓
	E			✓	✓	✓	✓	✓		✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
	D	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	C	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSP430F2111	I						✓			✓			✓	✓	✓		✓	✓	✓	✓	✓		✓				✓	✓	✓
	H						✓	✓		✓			✓	✓	✓		✓	✓	✓	✓	✓		✓				✓	✓	✓
	G			✓	✓	✓	✓	✓		✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓
	E			✓	✓	✓	✓	✓		✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
	D	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	C	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSP430F2121	I						✓			✓			✓	✓	✓		✓	✓	✓	✓	✓		✓				✓	✓	✓
	H						✓	✓		✓			✓	✓	✓		✓	✓	✓	✓	✓		✓				✓	✓	✓
	G			✓	✓	✓	✓	✓		✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓
	E			✓	✓	✓	✓	✓		✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓
	D	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	C	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSP430F2131	I						✓			✓			✓	✓	✓		✓	✓	✓	✓	✓		✓				✓	✓	✓
	H						✓	✓		✓			✓	✓	✓		✓	✓	✓	✓	✓		✓				✓	✓	✓
	G			✓	✓	✓	✓	✓		✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓
	E			✓	✓	✓	✓	✓		✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓
	D	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	C	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Detailed Bug Description

BCL6 BCL6 - Bug description

Module: BasicClock, Function: DCO calibration error

The DCO calibration values stored in the information memory may differ by up to ±2%.
This applies to all Revision C and the following Lot-Trace code Revision D devices:

- Lot#57FNFCK
- Lot#58E081K
- Lot#58E521K
- Lot#58F5K9K
- Lot#58F5KTK
- Lot#58F860K
- Lot#58F866K
- Lot#58FKL3K
- Lot#59C279K

Workaround:
Recalibrate the DCO using a reference clock source.

Detailed Bug Description (continued)

BCL8 BCL8 - Bug description

Module: BasicClock, Function: Erroneous ISR DCO DC generator enable

When MCLK and SMCLK are sourced from a clock other than the DCO, the SCG0 DCO control bit is erroneously cleared upon interrupt service routine entry from LPMx by the CPU. This enables the DCO dc generator causing additional current consumption. After executing the RETI instruction, the status register SCG0 bit returns to a set state turning off the dc generator, eliminating this added current. The increased current occurs only during the interrupt service routine, and is in the range of 20 μA at 3 V.

Workaround:
Set SCG0 in software at the beginning of an interrupt service routine. This reduces the time that the dc generator is enabled to four MCLK cycles.

BCL9 BCL9 - Bug description

Module: BasicClock, Function: ACLK divider modifications require delay before entering LPM3

After modifying the DIVAx bits, immediately entering LPM3 can cause the modification to be ignored and the divider settings not to take effect. Reading back the DIVAx bits indicates the intended setting, even when the divider has not been correctly applied.

Workaround:
When the DIVAx bits are modified, a delay of one complete ACLK (VLO or LFXT1CLK) period must elapse before entering LPM3. The delay is necessary only the first time LPM3 is entered after the DIVAx bits are modified. After the one-period delay, LPM3 may be entered and exited normally without additional delays.

BCL10 BCL10 - Bug description

Module: BasicClock, Function: MCLK = ACLK and P2SEL control bits

When using ACLK as the CPU MCLK clock source, the oscillator failsafe feature does not automatically switch MCLK to the DCO if the P2SEL6 or P2SEL7 bits are cleared. This applies when ACLK = LFXT1 (e.g., external low-frequency clock source). The CPU halts operation, because no MCLK signal is present.

Workaround:
None

BCL11 BCL11 - Bug description

Module: BasicClock, Function: Watchdog failsafe when using ACLK

When using ACLK as the WDT+ clock source, the WDT+ oscillator failsafe feature does not automatically switch to the DCO if the P2SEL6 or P2SEL7 bits are cleared. This applies when ACLK = LFXT1 (e.g., external low-frequency clock source). The WDT+ halts operation, because no clock signal is present.

Workaround:
None

Detailed Bug Description (continued)

BSL5 BSL5 - Bug description

Module: Bootstrap Loader

If the RST/NMI pin is configured to NMI, the bootstrap loader may not be started. Unpredictable operations results.

Workaround:
None

CPU5 CPU5 - Bug description

Module: Interrupt Handler: Incorrect interrupt vector fetch

When an interrupt is accepted and the interrupt vector is applied to the MAB, an interrupt vector address 32 bytes lower than the expected address may be applied to the address bus. This is true only when the stack pointer address is in one of the following ranges:
0x02C0 to 0x02DF
0x0280 to 0x029F
0x0240 to 0x025F
0x0200 to 0x021F

Workaround:
Solution #1: Limit stack depth to 32 bytes.
Solution #2 (Recommended, works under all conditions): Copy the interrupt vector table (0xFFE0 to 0xFFFF) in flash to addresses 0xFFC0 to 0xFFDF, respectively.

Detailed Bug Description (continued)

CPU6 CPU6 - Bug description

Module: CPU, incorrect ADD instruction

When the CPU executes an ADD (.B or .W) instruction using indirect addressing mode with destination R1 or R4 to R15, directly after a RET or RETI instruction, the addition is executed twice. This bug does not apply in the case that the indirect source working register is R2 or R3 (constant generator access for 2 or 4 are valid).

The instruction word mask that corresponds to the CPU6 condition is as follows:

Format I Instruction
(See MSP430x2xx Users Guide)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
op-code				**s-reg**				**Ad**	**B/W**	**As**		**d-reg**			
MSB															**LSB**
0	**1**	**0**	**1**	**X**	**X**	**X**	**X**	**0**	**X**	**1**	**0**	**X**	**X**	**X**	**X**

"X" is "Don't Care"
s-reg = R2 or R3 are not effected
d-reg = R0, R2 or R3 are not effected

Workaround:
1 (Preferred): Use indexed addressing with offset 0 instead of indirect addressing:
ADD(.B/.W) 0(R10),R11
Note that some assemblers may convert this instruction to indirect addressing.

Alternatively, when the source is in assembly, making the following changes works around the bug. Perform the following only if workaround 1 is not an available option.
2A: Place an instruction (e.g. NOP) between any instances of CALL and ADD(.B/.W) instructions where the ADD immediately follows the CALL.

2B: Replace all RETI instructions with 'DW 01301h'. The instruction word for RETI is 01300h. By replacing this directly in assembly with 01301h, the bug is avoided and the RETI instruction is executed properly. For example:

```
WDT_ISR                     ; Exit LPM3 on reti
        bic.w #LPM3,0(SP)   ;
        ;reti               ; Replace RETI with
        DW  01301h          ; this line of code
```

Please contact Texas Instruments for questions regarding CPU6.

Detailed Bug Description (continued)

CPU14 CPU14 - Bug description

Module: CPU, Function: Erroneous setting of SCG0 after reset

The SCG0 bit in the CPU status register (SR) is set after any reset (PUC or POR) if bit 6 in the reset vector destination address is set. Setting SCG0 turns off the DCO DdcC generator when DCOCLK is not used for MCLK or SMCLK.

Workarounds:
1) As the error only occurs after PUC or POR, it is sufficient to clear the SCG0 bit at the beginning of the program code. Example:
   bic.w #SCG0, SR
2) Avoid using reset destination addresses where bit #6 is set. Allowed reset vector destination addresses are: xx0xh, xx1xh, xx2xh, xx3xh, xx8xh, xx9xh, xxAxh, xxBxh.

When any of the above reset destination addresses are used, SCG0 is valid.

FLASH22 FLASH22 - Bug description

Module: Flash, Function: Flash controller may prevent correct LPM entry

When ACLK (or SMCLK) is used as the flash controller clock source, and this clock source is deactivated due to a low-power mode entry while a flash erase or write operating is pending, the flash controller keeps ACLK (or SMCLK) active even after the flash operation has been completed. This results in an incorrect LPM entry and increased current consumption. Note that this issue can occur only when the flash operation and the low-power mode entry are initiated from code located in RAM.

Workaround:
Do not enter low-power modes while flash erase or write operations are active. Wait for the operation to be completed before entering a low-power mode.

JTAG15 JTAG15 - Bug description

Module: JTAG, Function: Using TDO as input for TDI does not work when JTAG is used to drive the MAB

When the TDO is used as TDI, the MAB is not correctly driven by JTAG. This causes problems when using the function GangProgramTarget() of the GANG430.DLL. This function is usually used for programming individual serial numbers, for programming calibration data into devices, and for restoration of DCO calibration data after erasing of Segment A.

Workaround:
Upgrade to GANG430.DLL version 1.42 or higher.

PORT8 Port8 - Bug description

Module: PORT2, Function: Increased leakage current

The LFXT1 oscillator circuit is not disabled when P2SEL.6 = 0 or P2SEL.7 = 0. This can result in a higher leakage current on P2.7 XOUT when P2DIR.7 = 0 and P2SEL.7 = 0.

Workaround:
If using P2.7 as an input, select the bypass mode by setting the LFXT1Sx = 11

Detailed Bug Description (continued)

PORT10 Port10 - Bug description

Module: Digital I/O, Function: Pullup/pulldown resistor selection when module pin function is selected

When the pullup/pulldown resistor for a certain port pin is enabled (PxREN.y = 1) and the module port pin function is selected (PxSEL.y = 1), the pullup/pulldown resistor configuration of this pin is controlled by the respective module output signal (Module X OUT), instead of the port output register (PxOUT.y).

Workaround:
None. Do not set PxSEL.y and PxREN.y at the same time.

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
Clocks and Timers	www.ti.com/clocks	Digital Control	www.ti.com/digitalcontrol
Interface	interface.ti.com	Medical	www.ti.com/medical
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
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

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