MSP430F47x3/F47x4 Device Erratasheet

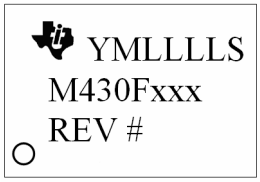
Current Version

Devices	Rev:	COMP3	FLASH19	FLASH24	FLASH25	SDA4	TA12	TA16	TB2	TB16	USCI13	USCI19	USCI20	USCI21	XOSC5
MSP430F4783	H	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	I		✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSP430F4784	H	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	I		✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSP430F4793	H	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	I		✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSP430F4794	H	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	I		✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Note: See Appendix for prior revisions

Package Markings

PZ100: LQFP (PZ) 100 pin



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

Detailed Bug Description

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

FLASH24 FLASH24 - Bug description

Module: Flash, Function: 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.

Detailed Bug Description (continued)

SDA4 SDA4 - Bug description

Module: SD16_A, Function: Reduced SINAD performance at certain input voltage levels

The performance of the SD16_A maybe degraded due to reduced SINAD when the level of the analog input is between 20 mV and 120 mV. This can occur on any channel, irrespective of their PGA settings.

Workaround:
1) Avoid the use of any PGA settings less than 16 with the common mode voltage of zero, that most likely accommodate input signal levels that fall under this range

Or

2) Introduce a common mode voltage, such as internal reference voltage of 1.2 V, to ensure that the input signal level is outside the range of 20 mV to 120 mV.

The following table shows the SD16_A performance with common mode voltage of zero:

SD16_A, performance (f_{SD16} = 1MHz, SD16OSRx = 256, SD16REFON = 1)

PARAMETER		TEST CONDITIONS		V_{CC}	MIN	TYP	MAX	UNIT
SINAD	Signal-to-Noise + Distortion Ratio	SD16GAINx = 1, Signal Amplitude V_{PP} = 500mV	f_{IN} = 50Hz, 100Hz see Note 1	3 V	82	84		dB
		SD16GAINx = 2, Signal Amplitude V_{PP} = 250mV		3 V	79	82		
		SD16GAINx = 4, Signal Amplitude V_{PP} = 125mV		3 V	64	76		
		SD16GAINx = 8, Signal Amplitude V_{PP} = 62mV		3 V	60	74		
		SD16GAINx = 16, Signal Amplitude V_{PP} = 31mV		3 V	66	70		
		SD16GAINx = 32, Signal Amplitude V_{PP} = 15mV		3 V	62	65		

NOTE: 1. The following voltages were applied to the SD16 inputs:
$V_{IN,A+}(t) = 0\ V + V_{PP}/2 \times \sin(2\pi \times f_{IN} \times t)$
$V_{IN,A-}(t) = 0\ V - V_{PP}/2 \times \sin(2\pi \times f_{IN} \times t)$
resulting in a differential voltage of $V_{diff} = V_{IN,A+}(t) - V_{IN,A-}(t) = V_{PP} \times \sin(2\pi \times f_{IN} \times t)$

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 gets lost.

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

Detailed Bug Description (continued)

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

TB2 TB2 - Bug description

Module: Timer_B, Interrupt is lost (slow ACLK)

Timer_B counter is running with slow clock (external TBCLK 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 TBR = CCRx).
Due to the fast MCLK the CCRx register increment (CCRx = CCRx + 1) happens before the Timer_B counter has incremented again. Therefore the next compare interrupt should happen at once with the next Timer_B counter increment (if TBR = 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.

TB16 TB16 - Bug description

Module: Timer_B, Function: First increment of TBR erroneous when IDx > 00

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

Workaround:
None

Detailed Bug Description (continued)

USCI13 USCI13 - Bug description

Module: USCI, UART/IrDA transmission FSM issue

When the USCI is in UART mode and its clock source is asynchronous to the CPU clock, writing to the transmit buffer during the stop-bit transmission of the previous character can cause unpredictable operation.

Workaround:
1) Use SMCLK, synchronous to MCLK, for the USCI clock source.
Or
2) Use the UCBUSY flag to determine if the USCI is idle before writing to the transmit buffer.
Or
3) Use application-specific timing to determine if the transmit buffer is safely written. For example, if using a 2400-Baud UART clocked from a 32768-Hz ACLK source to transmit two bytes from the UART, when the CPU is running from the DCO at 1 MHz, the second byte to be transmitted can be written to the transmit buffer as soon as the UCA0TXIFG flag becomes set, but must be written before the stop bit of the first byte. This application-dependent scenario can be safe, because the CPU clock is much faster than the USCI source clock and the baud clock such that it may be possible within the application to assure the CPU writes the second byte before the stop bit of the first byte is transmitted.

USCI19 USCI19 - Bug description

Module: USCI, Function: LPM4 may affect USCI operation

When SMCLK is used as the USCI clock source, and SMCLK gets deactivated due to a LPM4 entry, ongoing SPI master, I2C master, and UART transmit transaction will be interrupted. Also, while in LPM4, UART receive operation is nonfunctional.

Workaround:
Do not enter LPM4 while SPI master, I2C master, or UART transmit operations are active. Wait for the operation to be completed prior entering LPM4, or enter a different low-power mode instead. Also, do not use LPM4 in case of UART receive operation. Instead, use a different low-power mode.

Detailed Bug Description (continued)

USCI20 USCI20 - Bug description

Module: USCI, Function: I2C Mode Multi-master transmitter issue

When configured for I2C master-transmitter mode, and used in a multi-master environment, the USCI module can cause unpredictable bus behavior if all of the following four conditions are true:

1) Two masters are generating SCL
And
2) The slave is stretching the SCL low phase of an ACK period while outputting NACK on SDA
And
3) The slave drives ACK on SDA after the USCI has already released SCL, and then the SCL bus line gets released
And
4) The transmit buffer has not been loaded before the other master continues communication by driving SCL low

The USCI remains in the SCL high phase until the transmit buffer is written. After the transmit buffer has been written, the USCI interferes with the current bus activity and may cause unpredictable bus behavior.

Workaround:
1) Ensure that slave does not stretch the SCL low phase of an ACK period
Or
2) Ensure that the transmit buffer is loaded in time
Or
3) Do not use the multi-master transmitter mode

USCI21 USCI21 - Bug description

Module: USCI, Function: UART IrDA Receiver Mode

IrDA reception does not function correctly at certain baud rates. This occurs only when the IrDA Receive Filter via the UCAxIRRCTL register is enabled and the USCI source clock (BRCLK) is higher than 6 MHz.

Workaround:
1) Set the filter length (UCIRRXFLx in UCAxIRRCTL) to the maximum value 0x3F to achieve proper functionality.
Or
2) Reduce the frequency of the USCI source clock (BRCLK) to the IrDA module to be less than 6 MHz.

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

Current Version

Devices	Rev:	COMP3	FLASH19	FLASH24	FLASH25	SDA4	TA12	TA16	TB2	TB16	USCI13	USCI19	USCI20	USCI21	XOSC5
MSP430F4783	H	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSP430F4784	H	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSP430F4793	H	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MSP430F4794	H	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Detailed Bug Description

COMP3 COMP3 - Bug description

Module: Comparator_A: Control bits for comparator analog inputs not functional.

Bits P2CA0 and P2CA1 in CACTL2 used to connect the CA0 and CA1 device pins to the comparator module, respectively, have no effect when set to 1. The external connections to the comparator inputs are controlled instead by the CAPDx bits for the respective CA0 and CA1 port pins. (By definition, the CAPDx bits are used to disable the GPIO logic for each associated port pin.)

Workaround:
The CA0 and CA1 external inputs can be connected to the comparator internally by setting the bits CAPD6 = 1 and CAPD7 = 1 respectively. To disconnect an external analog signal from the CA0 or CA1 comparator inputs, CAPD6 or CAPD7 are cleared. Additionally in this case, the GPIO logic is no longer disabled and an external analog signal at the port input may cause increased current consumption through the digital input structure on the order of 10's of uA; the input leakage current is not affected.

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

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 guaranteed. 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.

FLASH25 FLASH25 - Bug description

Module: Flash, Function: Marginal Read Mode is not functional

The control bits for marginal read mode contained in the FCTL4 register are automatically cleared by any flash access. This prevents the marginal read mode from being used.

Workaround:
None

Detailed Bug Description (continued)

SDA4 SDA4 - Bug description

Module: SD16_A, Function: Reduced SINAD performance at certain input voltage levels.

The performance of the SD16_A maybe degraded due to reduced SINAD when the level of the analog input is between 20mV and 120mV. This can occur on any channel, irrespective of their PGA settings.

Workaround:
2) Avoid the usage of any PGA settings less than 16 with the common mode voltage of zero, that most likely accommodate input signal levels that fall under this range,
Or
2) Introduce a common mode voltage, such as internal reference voltage of 1.2V, to ensure that the input signal level is outside the range of 20mV to 120mV.
The following table shows the SD16_A performance with common mode voltage of zero:

SD16_A, performance (f_{SD16} = 1MHz, SD16OSRx = 256, SD16REFON = 1)

PARAMETER		TEST CONDITIONS		V_{CC}	MIN	TYP	MAX	UNIT
SINAD	Signal-to-Noise + Distortion Ratio	SD16GAINx = 1, Signal Amplitude V_{PP} = 500mV	f_{IN} = 50Hz, 100Hz see Note 1	3 V	82	84		dB
		SD16GAINx = 2, Signal Amplitude V_{PP} = 250mV		3 V	79	82		
		SD16GAINx = 4, Signal Amplitude V_{PP} = 125mV		3 V	64	76		
		SD16GAINx = 8, Signal Amplitude V_{PP} = 62mV		3 V	60	74		
		SD16GAINx = 16, Signal Amplitude V_{PP} = 31mV		3 V	66	70		
		SD16GAINx = 32, Signal Amplitude V_{PP} = 15mV		3 V	62	65		

NOTE: 1. The following voltages were applied to the SD16 inputs:
$V_{IN,A+}(t) = 0V + V_{PP}/2 \times \sin(2\pi \times f_{IN} \times t)$
$V_{IN,A-}(t) = 0V - V_{PP}/2 \times \sin(2\pi \times f_{IN} \times t)$
resulting in a differential voltage of $V_{diff} = V_{IN,A+}(t) - V_{IN,A-}(t) = V_{PP} \times \sin(2\pi \times f_{IN} \times t)$

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 gets lost.

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

Detailed Bug Description (continued)

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

TB2 TB2 - Bug description

Module: Timer_B, Interrupt is lost (slow ACLK)

Timer_B counter is running with slow clock (external TBCLK 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 TBR = CCRx).
Due to the fast MCLK, the CCRx register increment (CCRx = CCRx + 1) happens before the Timer_B counter has incremented again. Therefore, the next compare interrupt should happen at once with the next Timer_B counter increment (if TBR = CCRx + 1). This interrupt gets lost.

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

TB16 TB16 - Bug description

Module: Timer_B, Function: First increment of TBR erroneous when IDx > 00

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

Workaround:
None

Detailed Bug Description (continued)

USCI13 USCI13 - Bug description

Module: USCI, UART/IrDA transmission FSM issue

When the USCI is in UART mode and its clock source is asynchronous to the CPU clock, writing to the transmit buffer during the stop-bit transmission of the previous character can cause unpredictable operation.

Workaround:
1) Use SMCLK, synchronous to MCLK, for the USCI clock source.
Or
2) Use the UCBUSY flag to determine if the USCI is idle before writing to the transmit buffer.
Or
3) Use application-specific timing to determine if the transmit buffer is safely written. For example, if using a 2400-Baud UART, clocked from a 32768-Hz ACLK source to transmit two bytes from the UART, when the CPU is running from the DCO at 1 MHz, the second byte to be transmitted can be written to the transmit buffer as soon as the UCA0TXIFG flag becomes set, but must be written before the stop bit of the first byte. This application-dependent scenario can be safe, because the CPU clock is much faster than the USCI source clock and the baud clock such that it may be possible within the application to assure the CPU writes the second byte before the stop bit of the first byte is transmitted.

USCI19 USCI19 - Bug description

Module: USCI, Function: LPM4 may affect USCI operation

When SMCLK is used as the USCI clock source, and SMCLK gets deactivated due to a LPM4 entry, ongoing SPI master, I2C master, and UART transmit transaction will be interrupted. Also, while in LPM4, UART receive operation is non-functional.

Workaround:
Do not enter LPM4 while SPI master, I2C master, or UART transmit operations are active. Wait for the operation to be completed prior entering LPM4, or enter a different low-power mode instead. Also, do not use LPM4 in case of UART receive operation. Instead, use a different low-power mode.

Detailed Bug Description (continued)

USCI20 USCI20 - Bug description

Module: USCI, Function: I2C Mode Multi-master transmitter issue

When configured for I2C master-transmitter mode, and used in a multi-master environment, the USCI module can cause unpredictable bus behavior if all of the following four conditions are true:

1) Two masters are generating SCL
And
2) The slave is stretching the SCL low phase of an ACK period while outputting NACK on SDA
And
3) The slave drives ACK on SDA after the USCI has already released SCL, and then the SCL bus line gets released
And
4) The transmit buffer has not been loaded before the other master continues communication by driving SCL low

The USCI will remain in the SCL high phase until the transmit buffer is written. After the transmit buffer has been written, the USCI will interfere with the current bus activity and may cause unpredictable bus behavior.

Workaround:
1) Ensure that slave doesn't stretch the SCL low phase of an ACK period
Or
2) Ensure that the transmit buffer is loaded in time
Or
3) Do not use the multi-master transmitter mode

USCI21 USCI21 - Bug description

Module: USCI, Function: UART IrDA Receiver Mode

IrDA reception does not function correctly at certain baud rates. This occurs only when the IrDA Receive Filter via the UCAxIRRCTL register is enabled and the USCI source clock (BRCLK) is higher than 6 MHz.

Workaround:
1) Set the filter length (UCIRRXFLx in UCAxIRRCTL) to the maximum value 0x3F to achieve proper functionality.
Or
2) Reduce the frequency of the USCI source clock (BRCLK) to the IrDA module to be less than 6 MHz.

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

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