

Migrating from the MSP430F2xx and MSP430G2xx Families to the MSP430FR58xx/FR59xx/68xx/69xx Family

Priya Thanigai and William Goh

MSP430 Applications

ABSTRACT

This application report enables easy migration from MSP430F2xx flash-based MCUs to the MSP430FR58xx/FR59xx/68xx/69xx family of FRAM-based MCUs. For the migration guide to MSP430FR57xx, see [Migrating From the MSP430F2xx Family to the MSP430FR57xx Family](#). It covers programming, system, and peripheral considerations when migrating firmware. The intent is to highlight key differences between the two families. For more information on the use of the MSP430FR58xx/FR59xx/68xx/69xx devices, see the [MSP430FR58xx](#), [MSP430FR59xx](#), [MSP430FR68xx](#), and [MSP430FR69xx Family User's Guide](#). Although the MSP430F2xx and MSP430G2xx families are used as a base for comparison, similar considerations apply when migrating from MSP430F1xx and MSP430F4xx families.

Contents

1	Introduction	1
2	In-System Programming of Nonvolatile Memory	2
3	Hardware Migration Considerations	4
4	Device Calibration Information	4
5	Important Device Specifications	5
6	Core Architecture Considerations	5
7	Peripheral Considerations.....	10
8	Conclusion	15
9	References	15

List of Tables

1	Device Specifications	5
2	Comparison of FR59xx and F2xx Clock Systems.....	7
3	Comparison of Operating Modes and Wake-up Times	7
4	Comparison of Flash and FRAM on MSP430 MCUs	9
5	Comparison of USCI and eUSCI Modules	15

Trademarks

MSP430, EnergyTrace++, MSP430Ware, Grace are trademarks of Texas Instruments.
IAR Embedded Workbench is a trademark of IAR Systems.
All other trademarks are the property of their respective owners.

1 Introduction

The purpose of this application report is to highlight the key differences between the MSP430F2xx and MSP430G2xx families and the MSP430FR58xx/FR59xx/68xx/69xx family to ensure a smoother migration. It is divided into (1) system-level considerations such as power management, (2) changes when handling nonvolatile memory, and (3) peripheral modifications. With respect to the instruction set, the MSP430FR58xx/FR59xx/68xx/69xx family is completely backward code compatible with all other MSP430™ families. Any code migration is therefore impacted only by register or peripheral feature changes and slight variations in instruction cycle times, while the instruction set remains the same.

For any specific information, see the device-specific data sheet and errata.

NOTE: For the purpose of this application report, the term *F2xx* indicates the MSP430F2xx and MSP430G2xx families, and the term *FR59xx* indicates the MSP430FR58xx/FR59xx/68xx/69xx family.

2 In-System Programming of Nonvolatile Memory

2.1 Ferroelectric RAM (FRAM) Overview

Using FRAM nonvolatile memory is very similar to using static RAM (SRAM). FRAM's first introduction as an embedded memory in a general-purpose ultra-low power MCU was on Texas Instruments 16-bit MSP430 product line, the MSP430FR57xx family.

Some of the key attributes of FRAM are:

- FRAM is nonvolatile; that is, it retains its contents on loss of power.
- The embedded FRAM on MSP430 devices can be accessed (read or write) at up to a maximum speed of 8 MHz. Above 8 MHz, wait states are used when accessing FRAM.
- Writing to FRAM and reading from FRAM requires no setup or preparation such as pre-erase before write or unlocking of control registers (unless the MPU is used to protect the FRAM against write access).
- FRAM is not segmented and each bit is individually erasable, writable, and addressable.
- FRAM segments do not require an erase before a write.
- FRAM write accesses are low power, because writing to FRAM does not require a charge pump.
- FRAM writes can be performed across the full voltage range of the device.
- FRAM write speeds can reach up to 8 MBps with a typical write speed of approximately 2 MBps. The high speed of writes is inherent to the technology and aided by the elimination of the erase bottleneck that is prevalent in other nonvolatile memory technologies [8]. In comparison, typical MSP430 flash write speed including the erase time is approximately 14 kBps [7].

2.2 FRAM Cell

A single FRAM cell can be considered a dipole capacitor that consists of a film of ferroelectric material (ferroelectric crystal) between two electrode plates. Storing a 1 or 0 (writing to FRAM) simply requires polarizing the crystal in a specific direction using an electric field. This makes FRAM very fast, easy to write to, and capable of meeting high endurance requirements. Reading from FRAM requires applying an electric field across the capacitor similar to a write. Depending on the state of the crystal, it may be repolarized, thereby emitting a large induced charge. This charge is then compared to a known reference to estimate the state of the crystal. The stored data bit 1 or 0 is inferred from the induced charge. In the process of reading the data, the crystal that is polarized in the direction of the applied field loses its current state. Hence, every read must be accompanied by a write-back to restore the state of the memory location. With TI's MSP430 FRAM MCUs, this is inherent to the FRAM implementation and is completely transparent to the application. The write-back mechanism is also protected from power loss and completes safely under all power-fail events. The FR59xx power management system achieves this by isolating the FRAM power rails from the device supply rails in the event of a power loss. The FRAM power circuitry also uses built-in low dropout regulator (LDO) and a capacitor that store sufficient charge to complete the current write-back in the event of a power failure.

2.3 Protecting FRAM Using the Memory Protection Unit

Because FRAM is very easy to reprogram, it also makes it easy for erroneous code execution to unintentionally overwrite application code, just as it would if executing from RAM. To safeguard against erroneous overwriting of FRAM, a Memory Protection Unit (MPU) is provided. It is recommended to set up boundaries between code and data memory to increase code security and protect against accidental writes or erasures. The MPU allows users to separate blocks of FRAM and assign unique privileges to each block based on the application's requirement. For example, if a memory block is assigned 'read only' status, any write access to that block is prevented and an error is flagged. This is useful for storing constant data or application code that is not expected to change over the device lifetime. Code examples on how to configure the MPU are provided in the [MSP430FR5969 product folder](#).

2.3.1 Dynamically Partitioning FRAM

One unique property of FRAM is that when it is used in conjunction with the MPU, it provides the user the ability to dynamically shift the boundaries of code, data, and constant memory. This is done by setting up the MPU to establish read-only (constant), read-write only (variable), and read-execute only (code) segments. The resolution of each block is fixed to 4KB blocks [1].

See [MSP430 FRAM Technology – How To and Best Practices](#) on how to partition and enable the MPU.

2.4 FRAM Memory Wait States

The maximum FRAM memory access speed is 8 MHz. If the MCLK is operating faster than 8 MHz, wait-states are required to ensure reliable FRAM access. When using MCLK \geq 8 MHz, configure the FRAM wait states in software before configuring the MCLK frequency.

1. Configure the appropriate wait states.

```
FRCTL0 = FRCTLPW | NWAITS_x
```

2. Configure MCLK \geq 8 MHz.

For more information, see the *Wait State Control* section of the *FRAM Controller (FRCTRL)* chapter in [MSP430FR58xx, MSP430FR59xx, MSP430FR68xx, and MSP430FR69xx Family User's Guide](#).

2.5 Bootloader (BSL)

The BSL is software that is used to reprogram the MCU; for example, during field firmware updates. On the F2xx family of devices, the BSL uses a Timer_A-based UART and is located in ROM. The BSL is not erasable or customizable by the user.

The FR59xx family follows a similar approach wherein the BSL software is located in ROM. It occupies the address range 0x1000 to 0x17FF and cannot be erased or reprogrammed. In terms of the communication interface, the FR59xx is also based on the UART protocol similar to the F2xx devices. However, it uses the hardware USCI_A module to implement UART communication instead of using Timer_A. Hence, the module pins UCA0TXD and UCA0RXD are used for BSL communication.

Disabling the BSL is possible with the FR59xx family by programming a signature to the BSL signature location. This process is documented in the *SYS* chapter in the [MSP430FR58xx, MSP430FR59xx, MSP430FR68xx, and MSP430FR69xx Family User's Guide](#). Note that the location and length of the BSL signature is different from the F2xx family.

FR59xx device variants using an I²C hardware interface for the BSL are also available. However, for a given device variant, only the factory-configured interface can be used.

2.6 JTAG and Security

On F2xx devices, the JTAG port is secured by blowing a physical fuse on one of the JTAG lines by subjecting it to a high voltage through a special procedure. This action is irreversible, and further access to the device is only possible through the BSL.

On the FR59xx devices, the physical fuse has been replaced by a programmable JTAG fuse. Securing the device involves writing a specific signature to the JTAG signature location. When the fuse is programmed, access to the device is only possible through the BSL (using the BSL password). However, when the BSL password is supplied, it is possible to clear the JTAG fuse and make JTAG communication available once again. Hence, on the FR59xx devices, blowing the JTAG fuse is reversible if the BSL password is known.

The FR59xx family also provides an addition feature: JTAG lock with password. The password is located at FRAM location 0xFF88 and can be one to four words in length. To be able to access JTAG, the tool chain needs to first provide the password, following which JTAG access is granted. Any access with an incorrect password prevents JTAG access. When a password is verified, complete JTAG access is possible until the next BOR event.

When using the IAR Embedded Workbench™ IDE, the option for providing the JTAG password is described in the *IAR Embedded Workbench C-SPY Debugging Guide for MSP430 Microcontroller Family*, which is available in the IAR installation directory under \430\doc.

When using the Code Composer Studio IDE v5.2 or higher, this option is available by editing the MSP430Fxxx.CCXML file under Target Configurations in the Advanced Setup Section, Advanced target Configuration. The procedure is described in the [Code Composer Studio User's Guide for MSP430](#).

2.7 Production Programming

MSP-GANG430 does not support FR69xx. The [MSP-GANG](#) production programmer supersedes the MSP-GANG430 and supports FR69xx.

3 Hardware Migration Considerations

- For JTAG and SBW connections on the FR59xx devices, see the [MSP430 Hardware Tools User's Guide](#).
- The FR59xx devices provide an internal pull-up resistor on the reset line, which eliminates the need for an external reset resistor. For details, see the Special Function Register (SFR) (SFRRPCR) in [2].
- The FR59xx devices assign the BSL pins to P2.0 and P2.1. For the pin numbers based on a given package, see the device-specific data sheet.
- The FR59xx devices do not provide internal load capacitors on the LFXT oscillator as in the F2xx family. Hence, it is required to have external load capacitors if the LFXT oscillator is used.
- The FR59xx devices do not support a high-frequency clock source on the LFXT oscillator. If a high-frequency clock source must be used, it must be connected to the secondary crystal oscillator (XT2).

4 Device Calibration Information

Some F2xx devices provide a TLV structure that supplies calibration values for the DCO frequency, ADC reference, and internal temperature sensor. The TLV structure is stored in Information Memory segment A (Info A) and can be erased by you [3]. A mass erase of the device that occurs if an incorrect BSL password is supplied results in the erasure of the factory calibrated constants.

To prevent this, the TLV information on FR59xx devices is stored in protected FRAM where it cannot be erased by in-application or external access (such as through the BSL). For details on the location and access of the TLV, see the device-specific data sheet.

Also, the Info A segment is completely available for application use on the FR59xx devices.

The TLV in the FR59xx family includes an extra field to store a random number seed that is generated on a per device basis at production. It is useful for encryption and decryption algorithms.

5 Important Device Specifications

Table 1 shows important differences in device-level electrical specifications. [4] [5]

Table 1. Device Specifications

Parameter	FR59xx	F2xx
Supply voltage range	1.8 V to 3.6 V ⁽¹⁾	1.8 V to 3.6 V
Maximum system frequency, f_{SYSTEM}	16 MHz at $V_{\text{CC}} = 1.8 \text{ V}$	4 MHz at $V_{\text{CC}} = 1.8 \text{ V}$ 12 MHz at $V_{\text{CC}} = 2.7 \text{ V}$ 16 MHz at $V_{\text{CC}} = 3.3 \text{ V}$
Minimum supply voltage for nonvolatile memory programming	1.8 V	2.2 V
Minimum analog supply voltage for ADC operation	1.8 V	2.2 V

⁽¹⁾ The minimum operating voltage is dependent on the SVSH voltage levels.

Note that the most significant impact that migrating to an FR59xx device brings to your system is in terms of the power consumption. The FR59xx demonstrates significant improvement in active and standby power consumption, during both typical conditions and across the voltage and temperature range of the device. The same is also true for peripherals in the FR59xx family, such as the ADC12_B, which show significant power improvements when compared to their F2xx counterparts. For details on the power consumption for each peripheral and of the device in active and standby modes, see the device-specific data sheet.

6 Core Architecture Considerations

6.1 Power Management Module (PMM)

The F2xx family of devices use a single voltage rail to power the chip; that is, a single power rail supplies both the analog peripherals and the digital core on the chip. The FR59xx family uses a split voltage supply. The external voltage supply on the DV_{CC} pin is fed to an internal low-dropout voltage regulator (LDO) that supplies the CPU, memories, and digital modules, while AV_{CC} supplies the analog modules (see Figure 1).

The PMM manages all functions related to the core voltage and its supervision. Its primary functions are, first, to generate a supply voltage for the core logic and, second, to provide several mechanisms for the supervision of both the voltage supplied to the device (DV_{CC}) and the voltage generated for the core (V_{CORE}).

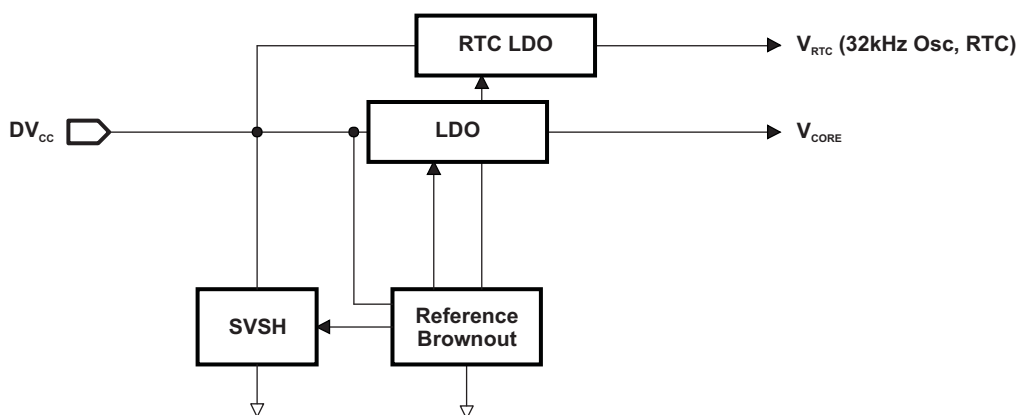


Figure 1. PMM Block Diagram

Using a split supply is especially advantageous as it allows the core to operate at a lower voltage, which brings significant power savings. It also ensures that the core receives a stable and regulated voltage over a wide supply range.

This allows the FR59xx devices to operate across the entire voltage range of the device at the maximum device frequency of 16 MHz. In comparison, F2xx devices have a relationship between system frequency and supply voltage that must be adhered to in order to ensure proper operation of the device.

Because supply voltage supervision is an important aspect of providing a stable supply or a notification in case of power failure, the FR59xx provides an high-side supply voltage supervision (SVSH) block. The SVSH handles the supervision of the external chip supply (DV_{CC}), the low-side supervision of the core is handled internally by the PMM.

The SVS threshold tracks directly with the device minimum supply of 1.8 V, and there is no need to program SVS high-side levels as in the F2xx family (in which the SVS feature is available on select devices only). Also, the SVSH block in the FR59xx is highly simplified. It is on by default at power-up and stays on in active, LPM0, LPM1, and LPM2 modes. It can be turned off in LPM3, LPM4, and LPMx.5 modes, if required.

A useful feature in the FR59xx family is the ability to trigger an NMI when the supply falls below the SVS level. This is useful, for example, if a battery-powered application is low on charge. In this case, an interrupt can be used to configure ports to consume the least power and set the device in LPM3.5, consuming approximately 500 nA to maximize battery life while preserving RTC for as long as possible. The interrupt can also be used to warn the end user about the low battery state and to power down gracefully.

One of the main differences between the two families that can be observed while debugging can be attributed to the PMM module. In the FR59xx family, the V_{CORE} regulator operates in two modes to conserve power: high-performance mode (used in active, LPM0, and LPM1 modes) and low-power mode (used in LPM2, LPM3, and LPM4 modes). When the FR59xx device is plugged into the debugger, it automatically forces the LDO to the high-performance mode regardless of the operating mode (active or LPM) that is set by the application code. In an application, this affects current consumption and wake-up times, which may cause the device to behave differently between stand-alone and debugger modes. When debugging with any of the lower LPMs (LPM2, LPM3, and LPM4), ensure that the debugger is disconnected to observe device performance accurately.

6.2 Clock System

The FR59xx Clock System (CS) shares some similarities with the F2xx Basic Clock System (BCS) in that it uses an internal digitally controlled oscillator (DCO) to provide pre-calibrated frequencies. The FR59xx also provides all of the same clock source options and system clocks as the F2xx family.

A significant difference in the FR59xx DCO is that it can be configured only to the factory-provided calibrated frequencies and does not provide the in-between frequency steps available on the F2xx DCO.

While the FR59xx can source MCLK at 16 MHz, it should be noted that FRAM access is limited to 8 MHz by the FRAM controller and wait states are required. For configuring wait states, see [Section 2.4](#). Code execution from RAM, accesses to peripherals, and DMA accesses between peripherals and RAM can be carried out at 16 MHz.

The ADC module's internal oscillator on the F2xx family has been renamed to MODOSC in the FR59xx family (similar to the F5xx family). This clock source is also used to derive a fail-safe clock source called LFMODCLK for the XT1 oscillator by using a divider.

The FR59xx CS supports the 'clocks-on-demand' feature. In the F2xx family, the availability of a system clock is impacted by entry into a low-power mode. For example, SMCLK is turned off in LPM3 and, hence, any peripheral such as a timer that uses SMCLK is inactive in LPM3. The FR59xx, however, allows the LPM settings to be overridden by a clock request. As long as there is an active request for a clock from a peripheral, the clock remains on, regardless of the LPM setting. This is most easily seen when there is increased power consumption when porting code between families. It is left to the user to disable any clock source requests that prevent the device from entering the required LPM. As an option, this feature can be disabled using the Clock System Control 6 Register (CSCTL6) (CLKREQEN bits).

EnergyTrace++™ Technology can be used to verify or diagnose if the clock sources are turned on or off as expected during LPMx. To learn more about EnergyTrace++ Technology, see [MSP430 Advanced Power Optimizations: ULP Advisor SW and EnergyTrace™ Technology](#).

[Table 2](#) lists important differences between the clock systems.

Table 2. Comparison of FR59xx and F2xx Clock Systems

Parameter or Feature	FR59xx	F2xx
Maximum system frequency, f_{SYSTEM}	16 MHz	16 MHz
DCO range	Calibrated frequencies only	0.06 to 26 MHz
Production calibrated frequencies	1 MHz, 2.66 MHz, 3.5 MHz, 4 MHz, 5.3 MHz, 7 MHz, 8 MHz, 16 MHz, 21 MHz, and 24 MHz	1 MHz, 8 MHz, 12 MHz, and 16 MHz
Clock sources for ACLK	LFXTCLK, VLOCLK, LFMODCLK (MODOSC / 128)	LFXTCLK, VLOCLK
LFMODCLK (MODOSC/128)	Available	Not Available
External crystal fail-safe options	XT1, LF: defaults to LFMODCLK XT2, HF: defaults to MODOSC	For any crystal failure: OFIFG is set, MCLK sourced by crystal defaults to DCO. Other clock sources do not have a fail-safe option.
Registers	CSCTL0 through CSCTL6	DCOCTL, BCSCTL1 through BCSCTL3
VLO control	Available with VLOOFF bit	Available with OSCOFF in LPM4
XT1 oscillator	Supports only LF mode	Supports LF and HF modes
XT2 oscillator	Supports up to 24 MHz	Supports up to 16 MHz
Internal load capacitors for XT1 oscillator	Not available	Available

6.3 Operating Modes, Wakeup, and Reset

Table 3 compares the operating modes that are available and the wake-up times from LPMs.

Table 3. Comparison of Operating Modes and Wake-up Times

Parameter or Feature	FR59xx	F2xx
LPM0, LPM1, LPM2, LPM3, LPM4	Available	Available
LPM3.5, LPM4.5	Available	Not available
Wake-up time from LPM0	$1.5 \times f_{\text{DCO}}$ $\approx 1.5 \mu\text{s}$ ($f_{\text{DCO}} = 1 \text{ MHz}$)	2 μs
Wake-up time from LPM1 or LPM2 ⁽¹⁾	6 μs	2 μs
Wake-up time from LPM3 or LPM4 ⁽¹⁾	7 μs	2 μs
Wake-up time from LPM3.5 ⁽¹⁾	250 μs	Not applicable
Wake-up time from BOR event ⁽¹⁾	1.5 ms (max)	2 ms (max)

⁽¹⁾ The values in this table are approximations; to find the values for a specific device, see the data sheet.

The code flow for entry into and exit out of low-power modes LPM0 to LPM4 remains the same in the FR59xx family as in the F2xx family. There are differences in functionality between the low-power modes on the F2xx compared to the FR59xx devices. These differences are described in the SYS chapter of the [MSP430FR58xx, MSP430FR59xx, MSP430FR68xx, and MSP430FR69xx Family User's Guide](#).

Two new low-power modes introduced in the FR59xx family are LPM3.5 and LPM4.5. In both modes, the V_{CORE} LDO is turned off, powering down the digital core, RAM, and peripherals. To wake up from LPM3.5, self-timed RTC interrupts or port interrupts are required. All other system interrupts are not available. The RTC module on the FR59xx device is powered from the V_{CC} rail and can, therefore, stay functional even when the core voltage is turned off. In LPM4.5, only port interrupts can be used to wake up the device.

It is important to understand that the LPMx.5 modes are inherently different from the typical LPMs (LPM0 through LPM4) in that a wake-up from these modes constitutes a device reset. Because RAM is not retained, the state of the application (if stored in variables located in RAM) and register initialization is lost. These modes are suited for applications that spend large amounts of time in 'deep sleep' and where wake-up time is not critical. Also, the frequency of wake up needs to be considered because there is an

energy penalty associated with the time spent during wake-up. For example, a 2xx application that wakes up every 1 ms from LPM3 to sample a signal can be more efficiently ported to LPM3 in FR59xx rather than LPM3.5. This is because LPM3.5 requires approximately 250 μ s to wake up, and the application will spend 25% of its duty cycle during wake-up, which significantly impacts the power gains that were achieved in moving from LPM3 to LPM3.5.

Consider a different application that wakes up once per minute to update a time stamp. In this case, LPM3.5 could be a better fit for maximizing power savings, because the average power during on time for LPM3.5 is 50% that of LPM3. Hence, selecting LPMs when migrating depends on the application and the on/off duty cycle that is required.

One important difference between the families is the behavior on reset. There are multiple levels of reset across all MSP430 families such as PUC, POR, and BOR. In the F2xx family, the program counter (PC) is reinitialized to the reset vector location on executing a PUC. In the case of a power cycle (POR), the PC is reinitialized after t_{DBOR} has elapsed [4]. In the FR59xx family, the behavior on executing a PUC is the same as the F2xx family as regards re-initialization of the PC and specific peripheral registers.

However, a deeper level of reset such as POR or BOR executes a boot code that is present in protected ROM. This boot code sets up the device and loads calibration settings that are essential to establish device functionality. Hence, the time to start up from a POR or BOR in the FR59xx family is different from the time in the F2xx family. For details, see the device-specific data sheet.

The FR59xx can also initiate all levels of reset in software (in the F2xx family, only a PUC can be initiated by software). The resets are initiated by setting the PMMSWBOR and PMMSWPOR bits in the PMMCTL0 control register.

6.4 Determining the Cause of Reset

In F2xx devices, a PUC can be triggered by multiple sources such as WDT timer expiration, WDT key violation, or flash key violation. To determine the cause of reset, it is necessary to investigate multiple registers, because each reset source is tracked by different interrupt flags and registers.

In the FR59xx devices, all sources of reset are combined into one System Reset Vector Register (SYSRSTIV), and it is no longer necessary to check multiple registers to determine the cause of reset. This register is very useful when debugging and lists all sources from all levels of reset (PUC, POR, and BOR).

6.5 Interrupt Vectors

The FR59xx devices use an interrupt vector (IV) structure for any interrupt service routine that is sourced by multiple flags.

For example, in the F2xx family, the USCI TX interrupt sources the RX and TX interrupt flags, and the USCI RX interrupt sources all the status flags. In the case of the FR59xx family, all of these interrupt flags are captured using a single interrupt vector UCBxIV. This allows interrupt servicing to be more efficient and ensures the same predefined latency for all interrupts.

6.6 FRAM and the FRAM Controller

6.6.1 Flash and FRAM Overview Comparison

The F2xx family flash controller is replaced by the FRAM controller in the FR59xx family.

The most significant differences between using FRAM and Flash pertain to (1) timing and (2) power requirements (see [Table 4](#)).

Table 4. Comparison of Flash and FRAM on MSP430 MCUs

Parameter	FRAM (FR5969) ⁽¹⁾	Flash (F2274) ⁽¹⁾
Program time for byte or word (max)	120 ns	116 μ s (approximately)
Erase time for segment (max)	Not applicable (pre-erase not required)	18 ms
Supply current during program (max)	No extra current during write (included in active power specification)	5 mA
Supply current during erase (max)	Not applicable (pre-erase not required)	7 mA
Nonvolatile memory maximum read frequency	8 MHz	16 MHz

⁽¹⁾ The values in this table are approximations; to find the values for a specific device, see the data sheet.

Because every read from an FRAM location is also a write, there is no excess current penalty due to write or erase. Hence, the power consumption when writing to a block of FRAM is the same as when reading from it. This is different from flash, where the writing process consumes excess power due to the operation of a device-internal charge pump. Similarly, FRAM does not require a pre-erase before write and is not segmented like flash. Hence, there is no added erase current (or erase time) when writing to FRAM.

In terms of the write time, FRAM is written in four-word blocks, and the write time is built into each read cycle. Hence, there is no difference between the read time and write time for an FRAM byte, word, or 4-word block. With regards to the read frequency, FRAM accesses (both read and write) are capped at 8 MHz. However, flash reads can take place at the maximum speed allowed by the device (f_{SYSTEM}), which is 16 MHz in the F2xx devices.

It should be noted that the speed of instruction execution in an FRAM-based system is affected by the architecture. The FR59xx uses a cache-based architecture that employs a combination of register and FRAM accesses when executing from nonvolatile memory. This allows the system throughput to be higher than the maximum allowable read frequency of 8 MHz.

6.6.2 Cache Architecture

The FRAM controller uses a 2-way associative cache that has a 64-bit line size. The cache stores prefetched instructions. The function of the FRAM controller is to prefetch four instruction words depending on the current PC location. The actual execution of these instructions is carried out in the cache. When the end of the cache buffer is reached, the FRAM controller preserves the four current words in the same cache line and fetches the next four words. If a code discontinuity is encountered at the end of a 2-way associative cache line, the cache is refreshed and the following four instruction words are retrieved from FRAM. However, if the application code loops back to a location already present in the cache when the last instruction in the cache is reached, the relevant instruction is simply executed directly from the cache instead of fetching code from FRAM again.

Note that only FRAM accesses are subject to the 8-MHz access limitation. When executing from cache, a system clock of up to 16 MHz can be used. Thus the cache is useful in (1) overcoming the 8-MHz limitation and increasing the average system throughput and (2) reducing overall active power by ensuring that most instructions are executed from it. Note that this is an instruction-only cache; all data is fetched directly from FRAM and is not cached.

The cached execution of instructions in the FR59xx family is different from the F2xx family, in which every instructions is directly executed from flash with no prefetches or caching, providing a 1:1 relationship between MCLK and instruction execution. For example, at MCLK = 16 MHz, eight two-cycle instructions can be executed in 16 clocks. For the FR59xx family, this relationship is application-dependent. The 1:1 relationship holds true only up to MCLK = 8 MHz. For MCLK > 8 MHz, the number of inserted wait states (directly proportional to how many times FRAM is accessed) determines the MCLK:instruction-execution ratio.

To provide another application example, with MCLK = 16 MHz, a JMP \$ instruction (single cycle) is executed at the same rate in both devices. This is because the FR59xx fetches this instruction once and stores it in cache where it can be executed at the maximum MCLK speed. However, a loop that has more than eight instruction words would require accessing the FRAM every time a cache refresh is needed. These FRAM accesses take place at MCLK / 2 = 8 MHz, thereby reducing the overall throughput of the system when compared to an F2xx device.

6.7 RAM Controller (RAMCTL)

MSP430x2xx MCUs do not have a RAM Controller. The RAM Controller allows reduction of the leakage current during LPM3 and LPM4. The RAM is partitioned in one to four sectors, depending on the device. See the device-specific data sheet for sector allocation and size. Each sector can be individually powered down in LPM3 and LPM4 to save leakage. Note that data is lost when sectors are powered down in LPM3 and LPM4.

The FR5962/4 and FR5992/4 MCUs have 8KB of SRAM. The SRAM is made up of three sectors. Sector 0 = 2KB, Sector 1 = 2KB, and Sector 2 = 4KB. The 4KB of Sector 2 is shared with the Low-Energy Accelerator (LEA) peripheral when the LEA module is activated. When the LEA module is not activated, Sector 2 SRAM can be used normally. See the device-specific data sheets for more information.

7 Peripheral Considerations

Some of the peripherals in the FR59xx family have new features or existing features that are implemented differently. This section highlights the peripheral differences.

7.1 Watchdog Timer

The main difference between the two families lies in the fail-safe operation.

In the F2xx family, WDT is typically timed by ACLK, which is sourced by a crystal or the VLO. If crystal failure occurs, WDT defaults to MCLK. If MCLK is also sourced by a crystal, the DCO is automatically activated.

In the FR59xx family, the WDT fail-safe defaults to VLO instead of the DCO.

7.2 Ports

7.2.1 Digital Input/Output

The main differences in the FR59xx general-purpose I/O (GPIO) pins are:

- All GPIOs have internal configurable pull-up and pull-down resistors.
- P3 and P4 ports are also interruptible in the FR59xx devices (only P1 and P2 in the F2xx devices)
- JTAG functionality in the FR59xx devices is multiplexed with GPIO pins on Port J
- Peripheral function select in the FR59xx devices uses two registers: Port x Function Selection Register 0 (PxSEL0) and Port x Function Selection Register 1 (PxSEL1). These two registers can be set or cleared simultaneously using the Port x Complement Selection Register (PxSELC) to avoid intermediate configurations.

NOTE: GPIOs will not function after reset until this change has been made to firmware.

By default, after BOR, all digital I/O are set as high-impedance with Schmitt triggers and their module functions disabled to prevent any cross current. This enables reduced power consumption when the device starts up.

The following code sequence is required to initialize GPIOs on reset and wakeup from LPMx.5:

1. Initialize all port pin registers as required for function: Port x Direction Register (PxDIR), Port x Pullup or Pulldown Resistor Enable Register (PxREN), Port x Output Register (PxOUT), Port x Select Register 0 (PxSEL0), Port x Select Register 1 (PxSEL1), and Port x Interrupt Edge Select Register (PxIES).
2. Clear the LOCKLPM5 bit.

```
PM5CTL0 &= ~LOCKLPM5
```
3. If not waking up from LPMx.5, clear all Port x Interrupt Flag Register (PxIFG) to avoid erroneous port interrupts.
4. Enable port interrupts using Interrupt Port x Interrupt Enable Register (PxIE).

For more information, see the *Configuration After Reset* section of the *Digital I/O* chapter in the [MSP430FR58xx, MSP430FR59xx, MSP430FR68xx, and MSP430FR69xx Family User's Guide](#).

7.2.2 Capacitive Touch I/O

The main difference in the capacitive touch implementation between the G2xx and the FR59xx lies in the selection of the port pins and the internal wiring of the capacitive touch I/O to the timer.

In the F2xx and G2xx devices with the pin oscillator feature, the selection of the pin oscillator is enabled through the PxSELY register.

On some G2xx devices, only one pin oscillator can be enabled at a time.

In the FR59xx devices, there are two registers CAPTIO0CTL and CAPTIO1CTL. Each of these registers can be used to select a port and a specific pin in that port that can then be used as a capacitive touch I/O.

For example, the CAPTIOPOSELx field in the Capacitive Touch IO x Control Register (CAPTIO0CTL) can select port 1. The CAPTIOISELx field in the same register can select pin 5. Hence, pin 1.5 is designated as a capacitive touch I/O.

For each CAPTIOxCTL register, the selected capacitive touch I/O is hard wired (internally connected) to a specific timer. According to the device-specific data sheet, CAPTIO0CTL selection is internally connected to TA2 and the CAPTIO1CTL selection is internally connected to TA3.

Hence, in the case of the FR59xx, it is possible to connect two capacitive touch elements to two different timers that can then be sampled simultaneously. Conceptually, the principle of generating a capacitive touch based oscillation that is then fed to a timer remains the same in both device families.

7.3 Analog-to-Digital Converters

7.3.1 ADC12 to ADC12_B

Some of the significant differences between the ADC12 and the ADC12_B module are:

- The ADC12_B is significantly lower power than the ADC12. To compare parameters such as supply current or reference buffer current, see the device-specific data sheet.
- The ADC12_B can operate across the entire voltage range of the device (1.8 V to 3.6 V).
- The ADC12_B module supports 16 single-ended external input channels. These can be combined to form 8 external differential input channels
- The ADC12_B provides a new feature called the Window Comparator that allows you to set threshold levels and compare conversion results to the thresholds. The thresholds are set in LSB units and, if a conversion result falls within the range of preset low and high threshold levels, an interrupt is triggered. Interrupts are also provided for when conversion results fall above or below the preset threshold levels. The same window comparator thresholds are shared among all channels, and conversion results from different channels are compared against the same threshold levels when the feature is enabled. This feature is extremely useful for saving power, because it allows the device to stay in LPM until the ADC input reaches a specific threshold. All other conversion results are discarded automatically, and the

device only wakes up on threshold triggers.

- The ADC12 requires 13 ADC12CLKs for sample conversion. In contrast, the ADC12_B requires 10, 12, or 14 ADC12CLKs for sample conversion at 8-bit, 10-bit, or 12-bit resolutions, respectively.
- The formula for calculating the minimum sample time as highlighted in the section *Sample Timing Considerations* has changed. For details, see the [MSP430FR58xx](#), [MSP430FR59xx](#), [MSP430FR68xx](#), and [MSP430FR69xx Family User's Guide](#).
- The ADC12_B has 32 ADC12MEMCTLx (memory control) registers for sequenced channel operation and 32 dedicated interrupts for reading the conversion memory.
- The ADC12_B provides an option to enable 8-bit, 10-bit, or 12-bit conversion based on the ADC12RES bit in the ADC12_B Control 2 Register (ADC12CTL2).
- The ADC12_B provides the option to read data in 2s-complement format by setting the ADC12DF bit in the ADC12CTL2 register.
- An option to save power can be enabled at sampling frequencies less than 50 ksps using the ADC12PWRMD bit in the ADC12CTL2 register.
- For all parametric comparisons such as linearity, offset, total unadjusted error, and timing, see the device-specific data sheets. For all functional specifications on the ADC12_B, see the [MSP430FR58xx](#), [MSP430FR59xx](#), [MSP430FR68xx](#), and [MSP430FR69xx Family User's Guide](#).
- ADC12IE and ADC12IFG registers in the F2xx family have been replaced by ADC12_B Interrupt Enable 0 Register (ADC12IER0), ADC12_B Interrupt Enable 1 Register (ADC12IER1), ADC12_B Interrupt Flag 0 Register (ADC12IFG0), and ADC12IFG1 registers in the FR59xx family. To compare the register changes between the modules, see the device-specific user's guides.

7.3.2 ADC10 to ADC12_B

While both the ADC10 and ADC12_B use the SAR-ADC architecture, there are many differences in how the modules implement key features such as setting the sampling frequency, reading the converted result, and handling grouped channels. All of the features listed as improvements in [Section 7.3.1](#) apply to this section as well. Attempting to highlight every difference between the two modules is beyond the scope of this application report. Hence, an attempt has been made to focus on the key differences between the modules.

- ADC12_B shows significant power improvements with both the ADC supply current and the reference current.
- ADC12_B on the FR59xx supports 8 differential and 16 single-ended external inputs
- ADC12_B has a dedicated memory control register for each input channel. This allows the user to set unique properties such as voltage reference input and provides a separate memory buffer for each channel of the ADC. For the ADC12_B on the FR59xx, 32 such memory control registers are provided. When a group of channels is sampled, the conversion results are stored sequentially and can be read after all the channels have completed sampling.
- The DTC module for the ADC10 is replaced by the DMA in the ADC12_B.
- All of the ADC12_B interrupts are handled using the ADC12IV.
- The reference control has been moved out of the ADC block on the ADC12_B (see [Section 7.4](#)).
- The internal battery measurement channel (AV_{CC}) has changed from measuring $1/2 AV_{CC}$ on the ADC10 to $2/3 AV_{CC}$ on the ADC12_B.
- For all parametric comparisons such as linearity, offset, total unadjusted error, and timing, see the device-specific data sheets. For all functional specifications on the ADC12_B, see the [MSP430FR58xx](#), [MSP430FR59xx](#), [MSP430FR68xx](#), and [MSP430FR69xx Family User's Guide](#).

7.4 REF_A Module

In F2xx devices, the ADC reference is controlled using REFON and SREFx bits in the ADC registers. The two options that can be selected for the internal reference are 1.5 V and 2.5 V.

In the FR59xx devices, the REF_A module controls the reference voltage selection. The available options for internal reference are 1.2 V, 2.0 V, and 2.5 V. The internal reference voltage is shared with all modules that require it; for example, the DAC and LCD.

Note that the reference settling time has been greatly reduced on the FR59xx devices. The MSP430F2619 device with ADC12 has a reference settling time of 17 ms. In comparison, the REF_A module has a settling time of only 75 μ s. This allows for greater flexibility and power savings when the ADC and REF are used intermittently.

The following additional features are available in the REF_A module:

- Status indicators for the reference generator have been added through the REFGENACT and the REFBGACT bits.
- The REFGENRDY bit can be queried to check if the REF_A has settled.
- The REFGENBUSY is useful to detect when it is safe to change the REF_A settings; for example, the settings must not be changed when the ADC is in the middle of an active conversion.

7.5 Comparator_A to Comparator_E

The Comparator_E module provides the following additional features:

- The comparator internal reference is sourced by the REF module.
- Two individual interrupts for selecting between falling and rising edge of the comparator output (CEIES.CECTL1).
- The output polarity of the comparator can be inverted (CEOUTPOL.CECTL1).
- The internal voltage reference can be routed externally on a comparator pin.
- The RC filter delay is selectable in software.
- A voltage hysteresis generator is available to generate hysteresis without using additional external components.
- The registers and control bit setting between the two families vary widely, and it is recommended to see the family user's guides [2] when porting firmware.

7.6 Hardware Multiplier (HWMPY32)

The FR59xx HWMPY32 module supports 8-bit, 16-bit, 24-bit, and 32-bit operations, which are fully utilized by the C compiler when building code for the FR59xx family. The module additionally supports fractional number mode and saturation mode; however, these features must be accessed by directly manipulating the hardware multiplier's memory-mapped control registers or through use of a suitable software library.

See [MSPMATHLIB](#), which is an accelerated floating point math library. This library can deliver faster computation for most commonly used math functions. To learn more, see [MSPMATHLIB: An Optimized MSP430 Library of Floating-Point Scalar Math Functions](#).

7.7 DMA Controller

The DMA controller allows movement of data from one memory address to another without CPU intervention. The main difference for the DMA between the MSP430x2xx and the MSP430FR5/6xx MCUs is the number of DMA channels available. FRAM MCUs that contain a DMA controller can have up to 8 DMA channels available.

In MSP430x2xx MCUs, the DMAONFETCH bit controls when the CPU is halted for a DMA transfer. In FR58xx, FR59xx, FR68xx, and FR69xx MCUs, this bit is now DWARMWDIS.

DMA trigger selection depends on the device. See the device-specific data sheet for the number of channels and the trigger assignments.

7.8 Low-Energy Accelerator (LEA) for Signal Processing

The LEA module is a hardware engine designed for operations that involve vector-based arithmetic and signal processing for the MSP430FR599x MCUs. Compared to using the CPU for these operations, the LEA module offers up to 19 times faster performance and up to 20 times less energy consumption when performing vector-based digital signal processing computations such as FIR or IIR filtering, correlation, and FFT calculations. The LEA module requires MCLK to be operational; therefore, the LEA module is enabled in active mode or LPM0. When the LEA module is used, data operations are performed and shared on 4KB of SRAM. The MSP CPU and the LEA module can run simultaneously and independently unless they access the same system RAM.

The LEA operations are abstracted within the Digital Signal Processing (DSP) Library for MSP MCUs for ease-of-use. DSP Library functions can be used on any MSP device; however, if the LEA module is available on the device, DSP Library automatically uses the LEA module for vector math operations.

Before using the DSP Library APIs, first specify the input and output memory locations by allocating an array in which the locations need to reside within the shared 4KB of LEA SRAM memory. For example, to perform 256-point complex FFT with the LEA module, the data input array consists of 256-word real and 256-word complex values which totals to 512 words (1024 bytes).

For more information on the LEA module, see the following links:

- [Low-Energy Accelerator \(LEA\) Frequently Asked Questions \(FAQ\)](#)
- [Benchmarking the Signal Processing Capabilities of the Low-Energy Accelerator](#)
- [Digital Signal Processing \(DSP\) Library for MSP MCUs](#)
- [MSP430FR5994 Product Page](#)
- [MSP430FR58xx, MSP430FR59xx, MSP430FR68xx, and MSP430FR69xx Family User's Guide](#)

7.9 Communication Modules

7.9.1 USI to eUSCI

The USI module that is available on some F2xx devices is architecturally different from the eUSCI module. The USI module is built primarily on a shift register that is used in conjunction with a counter to shift out data bits. Any protocol-specific aspects for SPI or I²C communication are implemented through software. Hence, it can be said that the implementation of the USI module is a combination of equal parts firmware and hardware.

In contrast, the eUSCI module is almost completely hardware based. The application firmware is only required to configure the module based on the protocol being used and then access interrupts to receive or transmit data. Hence, with regards to migrating firmware from the USI to the eUSCI, code cannot be reused. It is instead recommended to see the code examples provided online in the device product folder that show easy setup of the eUSCI module and handling of interrupts.

The eUSCI module handles all communication-specific implementation details in hardware, which allows the application to be better power-optimized and to service data transmission and reception more efficiently.

The USI module supports SPI and I²C protocols while the eUSCI module supports SPI, I²C, and UART protocols.

7.9.2 USCI to eUSCI

The architecture and the internal state machine of the eUSCI is very similar to the USCI module in the F2xx family. However, there are many new features added in the eUSCI as well as changes made to the existing features. While most of the code is still compatible, TI recommends reviewing the register names. [Table 5](#) shows most of the significant differences between the families. For more detailed information, see [Migrating from the USCI to the eUSCI Module](#).

Table 5. Comparison of USCI and eUSCI Modules

Parameter or Feature	USCI (F2xx)	eUSCI (FR59xx)
UART		
Enhanced baud rate generation	No	Yes
TXEPT interrupt (similar to USART)	No	Yes
Start edge interrupt	No	Yes
Selectable glitch filter	No	Yes
Interrupt vector generator	No	Yes
SPI		
Enhanced baud rate generation	No	Yes
Maximum baud rate	4 to 6 MHz	7 ⁽¹⁾
Interrupt vector generator	No	Yes
I²C		
Preload of transmit buffer	No	Yes
Clock low timeout	No	Yes
Byte counter	No	Yes
Multiple slave addressing	No	Yes
Address bit mask	No	Yes
Hardware clear of interrupt flags	Yes	No
Interrupt vector generator	No	Yes

⁽¹⁾ Calculated based on SPI timing with another MSP430FR5969 device in slave mode. For the formula to calculate the maximum baud rate, see device-specific data sheet.

8 Conclusion

This application report describes many of the key feature changes and new modules in the MSP430FR59xx family compared to the MSP430F2xx family. While this document is intended to be comprehensive, there may be minor differences between the F2xx and the FR59xx families that have not been covered here. For details of a given device, the device-specific data sheet is always the best source of information. For module functionality and use, see the [MSP430FR58xx](#), [MSP430FR59xx](#), [MSP430FR68xx](#), and [MSP430FR69xx Family User's Guide](#).

For developers working with the FR59xx devices, two useful resources for getting started include [MSP430Ware™ software](#) and [Grace™ graphical configuration tool](#), a plug-in for Code Composer Studio™ IDE.

9 References

1. [MSP430FR58xx, MSP430FR59xx, MSP430FR68xx, and MSP430FR69xx Family User's Guide](#)
2. [MSP430x2xx Family User's Guide](#)
3. [MSP430F22x4, MSP430F22x2 Mixed Signal Microcontroller](#)
4. [MSP430FR59xx, MSP430FR58xx Mixed Signal Microcontroller](#)
5. [Migrating from the USCI Module to the eUSCI Module](#)
6. [Maximizing FRAM Write Speed on the MSP430FR5739](#)
7. [Migrating From the MSP430F2xx Family to the MSP430FR57xx Family](#)
8. [Code Composer Studio™ User's Guide for MSP430™ User's Guide](#)
9. [MSP430 Hardware Tools User's Guide](#)

Revision History

Changes from March 30, 2016 to November 3, 2016	Page
• Removed list item that incorrectly stated that the internal battery measurement channel on the ADC12_B has changed to measure 2/3 AVCC	12
• Changed <i>LEA_SC</i> to <i>LEA</i> throughout document	14
• Added more links for additional information on LEA in Section 7.8, Low-Energy Accelerator (LEA) for Signal Processing	14

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/sampterm.htm) (<http://www.ti.com/sc/docs/sampterm.htm>).

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