Safety Manual for MSP430G2xx, MSP430F5xx, and MSP430FR57xx Devices in IEC 60730 Safety Applications

User's Guide

Literature Number: SLAU552A
January 2014–Revised January 2016

Contents

List of Figures

List of Tables

User's Guide
SLAU552A–January 2014–Revised January 2016

Safety Manual for MSP430G2xx, MSP430F5xx, and MSP430FR57xx in IEC 60730 Safety Applications

1 Introduction

The IEC 60730-1:2010 standard *Automatic electrical controls for household and similar use - Part 1: General requirements* addresses safety in appliances and equipment that use microelectronics hardware and related software. Manufacturers of these types of end products must take steps to ensure safe operation of their products in order to meet the IEC 60730 standard. Annex H of this standard covers the aspects most relevant to microcontrollers including the three classifications of safety integrity for automatic electrical control by software:

- **Class A** functions such as room thermostats, humidity controls, lighting controls, timers, and switches. These are distinguished by not being relied upon for the safety of the equipment.
- **Class B** functions such as thermal cut-offs are intended to prevent unsafe operation of appliances such as washing machines, dishwashers, dryers, refrigerators, freezers, cookers, and stoves.
- **Class C** functions are intended to prevent special hazards such as explosions. These include automatic burner controls and thermal cut-outs for closed unvented water heaters.

This document covers the Class B level of safety that is described in these specifications. For the current release of this document, all references are applicable to the IEC 60730-1:2010 standard.

1.1 About This Document

As a system and equipment manufacturer or designer, you are responsible to ensure that your systems (and any Texas Instruments hardware or software components incorporated in your systems) meet all applicable safety, regulatory, and system-level performance requirements. All application and safety related information in this document (including application descriptions, suggested safety measures, suggested TI products, and other materials) is provided for reference only. You understand and agree that your use of TI components in safety-critical applications is entirely at your risk; and that you (as buyer) agree to defend, indemnify, and hold harmless TI from any and all damages, claims, suits, or expense resulting from such use.

This document is a safety manual for the MSP430™ Self Test Library (STL) that is included in the MSP430 IEC 60730 Software Package. The MSP430 STL is validated for Texas Instruments MSP430G23xx, MSP430G24xx, MSP430G25xx, MSP430G2x44, MSP430G2x55, MSP430F5xx, and MSP430FR57xx ultra-low-power 16-bit microcontroller (MCU) product families. This safety manual introduces the MSP430 STL developed for the previously mentioned microcontroller's families and the subsystem-based architecture that matches most low-power and portable applications. The inherent safety features of the MSP430 devices in combination with the MSP430 STL function calls can enable existing and emerging systems that require IEC 60730 safety compliance.

This safety manual provides information that is needed by system developers to assist in the creation of an end product that complies with the IEC 60730 class of safety system using a supported MSP430 MCU. This document contains:

- List of MCU modules to be tested for Class B compliance of IEC 60730.
- Typical application firmware with IEC 60730 Safety Supervisory Functions
- Software design and development process of STL
- MSP430 IEC 60730 STL development and release process
- An overview of MSP430 MCU Value Line G2xx, F5xx, and FR57xx product
- Architecture
- MSP430 Compiler and Tools Development Process and Tracking

It is expected that the user of this document has a general familiarity with the MSP430 product family. More information can be found at http://www.ti.com/msp430. This document is intended to be used in conjunction with the device-specific data sheets, family user's guides, device errata sheets, and other documentation for the products under development. This organization of technical content is intended to simplify development, reduce duplication of content, and avoid confusion.

Finally, this safety manual is part of the collateral material in the MSP430 IEC 60730 Software Package for F5xx, FR57xx, G23xx, G24xx, G25xx, G2x44, and G2x55 devices, and it helps you to implement the functional safety routines that are provided in the software package. The software package includes:

- MSP430 IEC 60730 Software Package User's Guide
- MSP430 IEC 60730 Software Package API Guide
- BSD licensed MSP430 STL source code
- Example projects for each supported MSP430 family

1.2 MSP430 and its Application Sectors

MSP430 MCUs are 16-bit RISC-based mixed-signal processors designed specifically for ultra-low power. MSP430 MCUs are highly integrated and offer a wide range of high-performance analog and digital performance in addition to a flexible clocking system. In addition, the MSP430FR57xx family features Ferromagnetic Random Access Memory (FRAM) unified memory, which allows dynamic partitioning and memory access that is 100 times faster than flash memory.

Application sectors for MSP430 MCUs include:

- Consumer electronics
- Energy harvesting
- Intelligent sensing
- Portable instrumentation
- Utility metering

G2xx Value Line, F5xx, and FR57xx are part of the MSP430 family of MCUs. These devices are available with different memory footprints and peripheral selections. Product-specific nomenclature and its derivatives are explained in http://www.ti.com/lsds/ti/microcontroller/16-bit_msp430/getting-started.page#decoder.

2 IEC 60730-Class B MCU Safety Compliance Features

Table 1 summarizes Annex H Table H.11.12.7 in the IEC 60730-1:2010 standard. The table in Annex H lists all MCU submodules that might need to be considered for IEC 60730 compliance. The reference number shown under "Acceptable Measure" (for example, H.2.16.5) is a reference to the complete definition of the acceptable measure in the IEC 60730 standard. The last column in Table 1 provides information regarding availability of an Application Program Interface (API) in MSP430 STL. The acceptable measure implemented by the APIs is also shown in Table 1. For more information regarding APIs included in MSP430 STL, refer to the *MSP430 IEC 60730 Software Package User's Guide*.

Table 1. Safety Compliance Features

MCU Component		Fault and Error	MCU Components Required to be Tested for Class B Compliance in IEC 60730 and Availability of Safety Function in MSP430 STL Acceptable Measure	API Available in MSP430 STL?
1	CPU			
1.1	Registers	Stuck at	• Functional test (H.2.16.5) or • Periodic test (H.2.16.6) using: – **Static Memory test (H.2.19.6)** or – Word protection with single-bit parity (H.2.19.8.2)	YES
1.3	Program Counter	Stuck at	• Functional test (H.2.16.5) • Periodic test (H.2.16.6) • Independent time-slot monitoring (H.2.18.10.4) or • **Logical monitoring of the program sequence (H.2.18.10.2)**	YES
2	Interrupt Handling and execution	No Interrupt	• **Functional test (H.2.16.5)** • Independent time-slot monitoring (H.2.18.10.4)	NO(1)
3	Clock	Wrong frequency	• Functional test (H.2.16.5) • **Independent time-slot monitoring (H.2.18.10.4)**	YES
4	Memory			
4.1	Non volatile memory	Single bit fault	• **Periodic modified checksum (H.2.19.3.1)** • Multiple checksum (H.2.19.3.2) • Word protection with single-bit redundancy (H.2.19.8.2)	YES
4.2	Variable memory	Dynamic cross link	• **Periodic test (H.2.16.6)** • Word protection with single-bit redundancy (H.2.19.8.2)	YES
4.3	Addressing(2)	Stuck at		N/A
5	Internal data path(3)			
5.1	Data(3)	Stuck at		NO
5.2	Addressing(3)	Wrong address		NO
6	External Communication			
6.1	Data	Hamming distance	• Word protection with multi-bit redundancy including address (H.2.19.8.1) • CRC single word (H.2.19.4.1) • Transfer redundancy (H.2.18.2.2) • Protocol test (H.2.18.14)	NO
6.2	Addressing	Wrong address and multiple addressing	• Word protection with multi-bit redundancy including address (H.2.19.8.1) • CRC single word (H.2.19.4.1)	NO

(1) The MSP430 STL does not provide a function call. However, an example project in the software package shows how to perform the interrupt sweep in software.
(2) Memory tests indirectly check for stuck bits on the address bus. This function needs to be implemented by the user if external memory is connected to the MSP430.
(3) Memory tests indirectly provide some degree of internal data path coverage.

Table 1. Safety Compliance Features (continued)

MCU Component		Fault and Error	MCU Components Required to be Tested for Class B Compliance in IEC 60730 and Availability of Safety Function in MSP430 STL Acceptable Measure	API Available in MSP430 STL?
6.3	Timing	Wrong point in time and sequence	• Time-slot monitoring (H.2.18.10.4) • Schedule transmission (H.2.18.18) • Logical monitoring (H.2.18.15) • Reciprocal comparison (H.2.18.15) • Independent hardware comparator (H.2.18.3)	NO
7	Input/Output			
7.1	Digital IO	Incorrect value	**Plausibility check (H.2.18.13)**	YES
7.2.1	Analog IO	Incorrect value	**Plausibility check (H.2.18.13)**	YES
7.2.2	Analog mux	Wrong addressing	**Plausibility check (H.2.18.13)**	YES
9	Custom Logic			N/A

3 Typical Application Firmware With IEC 60730 Safety Supervisory Functions

This section describes the typical application firmware with IEC 60730 safety supervisory functions running on an MSP430 MCU. The example projects included in the MSP430 IEC 60730 software package follow the same execution sequence. Figure 1 highlights in blue the sections where the safety functions are executed in the application.

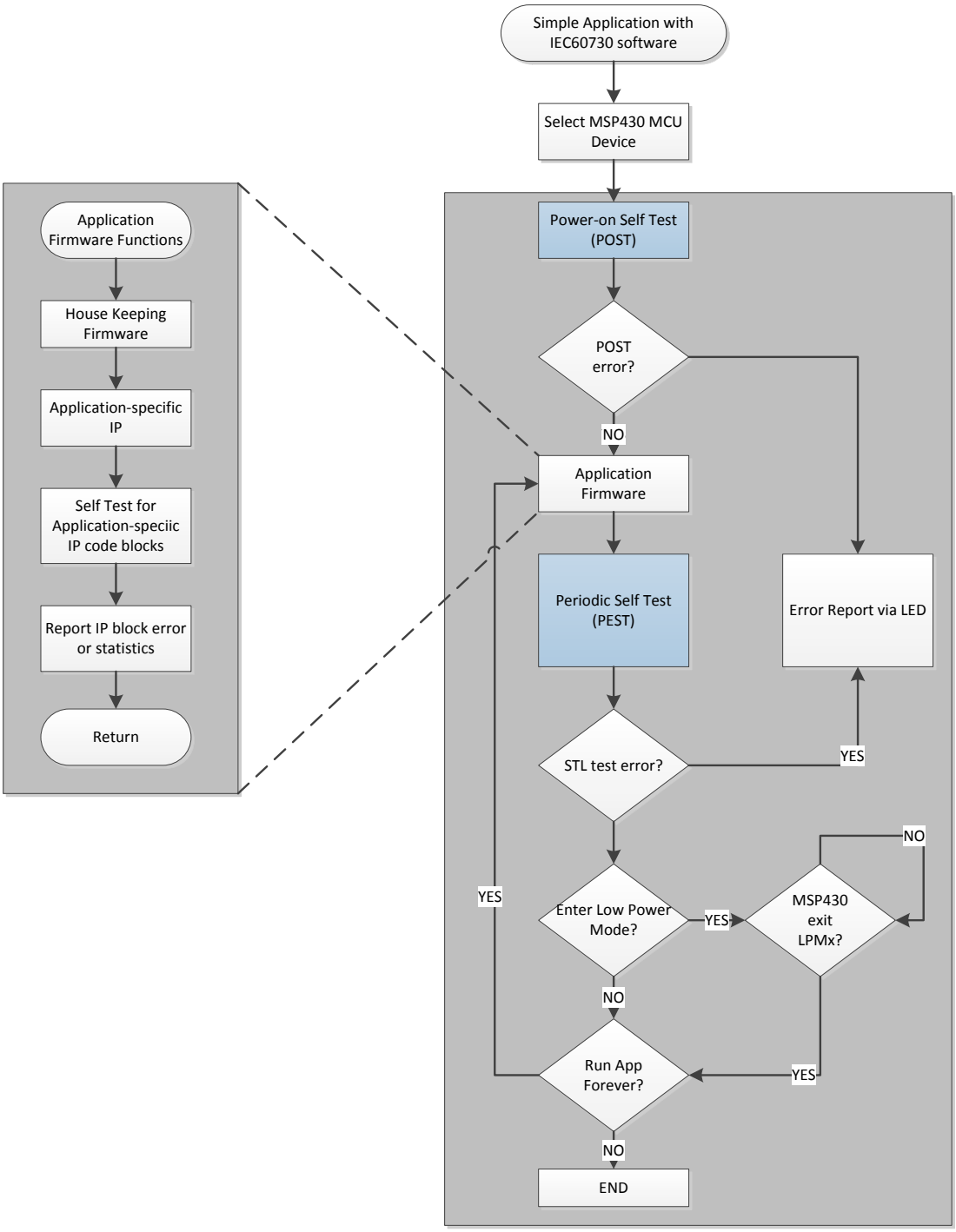

Figure 1. Typical Application Firmware Components With MSP430 IEC 60730 STL POST and PEST Functions

4 MSP430 IEC 60730 STL Development and Release Process

The design and test of the MSP430 IEC 60730 STL was developed with the elements of IEC 60730 "V-model" in mind. This method (see Figure 2) allows design, early validation, regression tests, and customization. This design method includes living working examples to explain code functionality and enables easy in-system validation.

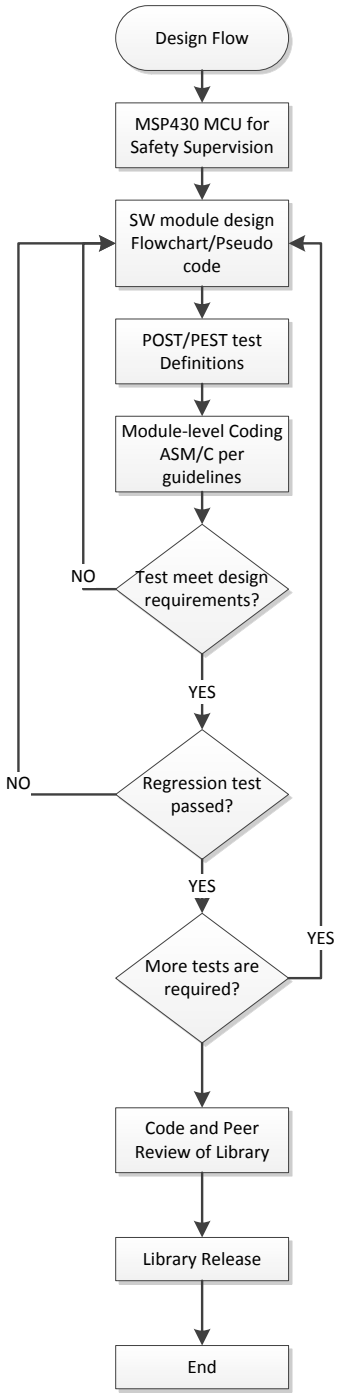

Figure 2. MSP430 IEC 60730 STL Design and Regression Flow

5 MSP430 MCU IEC 60730 Software Safety Development Process

The MSP430 STL APIs are written in MSP430 CPUX and CPUXv2 assembly and C (see Figure 3). Assembly functions are intended to do self test more efficiently and leverage architecture features. All of these functions are user-friendly functions that can be included in the end application. The following sections provide the highlights of coding rules and conventions used in the MSP430 STL.

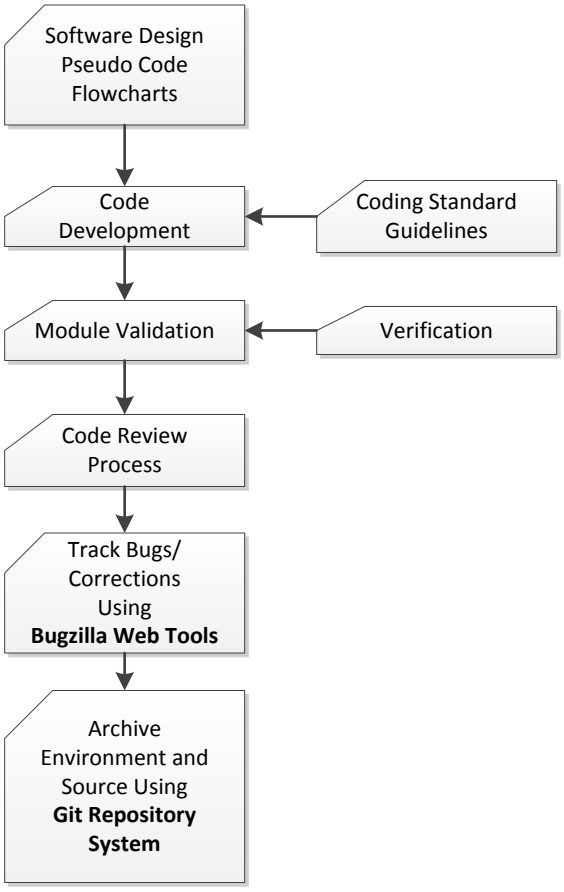

Figure 3. MSP430 STL Development Process

5.1 Software Design

Design of each module is documented through the use of a flowchart or pseudo code. The choice depends on the module and which best represents the functionality. The coding standards and naming conventions are described in Section 5.1.1.

5.1.1 Coding Standard and Naming Conventions

5.1.1.1 Variable Names

All variables are camel case starting with a lowercase letter. Global variables must begin with the letter 'g'.

```
uint16_t loopCounter
uint16_t gDataBuffer
```

Type definitions must begin with their module name in all capital letters, followed by an underscore ('_').

```
MODULE_FruitTypes
```

Defined constants must be in all capital letters and start with the module name.

```
#define MODULE_REGISTER_BIT (1 << 4)
```

5.1.1.2 Function Names

All functions must begin with their module name in all capital letters, followed by an underscore ('_').

Function names must be camel case starting with a lower-case letter.

UART_putChar();
PWM_setPeriod();
TIMER_getValue();

5.1.1.3 Comments

All comments must use the single-line comment delimiter "//" and must be indented to the same level as the code that they document.

5.1.1.4 Function Prototypes

All functions must be prototyped in a module's corresponding header file.

Each indent level is four spaces and all white space contains only spaces (no tabs).

Continued lines must be indented eight spaces.

5.1.1.5 Bracketed Statements

Bracketed statements must be in one of the following forms:

```
if (expr){
    stmt;
}else if (expr){
    stmt;
} else {
    stmt;
}
```

```
do {
    stmt;
} while (expr);
```

```
for (expr; expr; expr) {
    stmt;
}
```

```
switch (expr) {
    case CONST:
        break;
    case CONST:
        break;
    default:
        break;
    }
}
```

5.1.2 File Structure

All software in a file must reside in one of the following sections. The sections must appear in the following order within a file:

- Header, includes, defines, typedefs, globals, function prototypes
- For more details, see the header files in the software source in the IEC 60730 STL library.

5.1.3 Data Type

The following C99 data type must be used:

- bool for Boolean types
- (u)int_leastX_t (portable)
- (u)intX_t (architecture specific), where X is 8, 16, 32 or 64

5.1.4 Macro Definitions

Only single-line macros are allowed. Inline functions must be used to optimize functions.

5.1.5 Code Review Process

Peer-to-peer code review. All issues are entered in a Bugzilla (www.bugzilla.org) system for later resolution and tracking.

6 MSP430 Compiler and Tools Development Process and Tracking

Software code development and firmware debug is made using the Code Composer Studio™ tool set from Texas Instruments. Details of the tools and the environment are below. The Code Composer Studio tool set consists of three main components:

- Code Composer Studio IDE
- Code Generation Tools (Compiler)
- Debug Stack for MSP430 MCUs

These are covered in the following:

- **Code Composer Studio IDE and Related MSP430 Code Generation Tools (Compiler)**

  These products are developed inside TI's software development organization. All relevant information regarding the development process, tracking, and testing is provided on the external TI Development Tools Information wiki page: http://processors.wiki.ti.com/index.php/TI_Development_Tools_Information
- **MSP430 Debug Stack**

  This product is developed by TI's MSP430 development tools group. Similar to the Code Composer Studio IDE, these software releases are controlled and archived in compliance with TI's quality standards. The debug stack used inside Code Composer Studio is referred to as MSP430.DLLv3. All relevant information about this core component and related products is provided on the MSP Debug Stack wiki page: www.ti.com/mspds

7 MSP430 Architecture and Product Overview

The MSP430 16-bit microcontroller platform of ultra-low-power RISC mixed-signal microprocessors from Texas Instruments feature different sets of analog and digital peripherals targeted for various applications. These devices cater to a number of different end equipments including medical equipment, electricity and submetering, and home appliances such as smoke detectors and thermostats. The MSP430 G2xx Value Line, F5xx, and FR57xx families are part of this MSP430 microcontroller platform that can help enable IEC 60730 functional safety standard compliance in system applications including household appliances.

Figure 4 provides a very high-level overview of the MSP430 MCUs in the G2xx Value Line, F5xx, and FR57xx families.

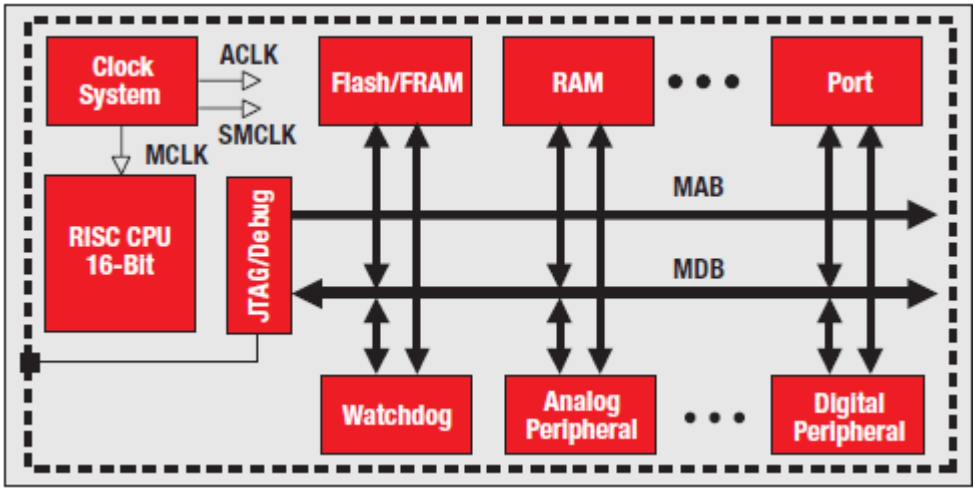

Figure 4. MSP430 MCU for IEC 60730 Applications – Overview of System Architecture

7.1 Targeted Applications and Product Safety Constraints

The MSP430 family of MCUs is targeted primarily for the ultra-low-power application space but is also well suited for a wide range of general-purpose applications. Target end equipments include portable medical and nonmedical devices; industrial and general-purpose sensing, control, and data-logging applications; utility metering; energy harvesting; and security applications.

As these devices are mass market products rather than custom products, a specific application implementation, configuration, and usage cannot be assumed. However, MSP430 MCUs are designed to meet most of the safety features mentioned in the IEC 60730 standard and, therefore, can help address various system-level safety needs. While the devices inherently have several component hardware level safety features, the application software is a critical part leveraging and building on the provided hardware features to meet functional safety compliance. A properly designed product using a MSP430 MCU should be capable of meeting functional safety requirements, if the designer uses the device within all of its stated data sheet specifications and implements the diagnostic hardware and software methods that are described in this document. System integrators should ensure that component and system safety concepts are assessed and adhered to during its development and product life cycle. For a list of best practices when developing safety applications, refer to Appendix A.

7.2 MSP430 MCU Architecture Classification

The following configurations of CPU, memory, and subsystems are available across many MSP430 G2xx, F5xx and FR57xx devices. In this document, subsystem refers to collection of digital and analog peripherals critical for system-on-chip (SOC) solutions. The above architecture breaks down in three types based on CPU architecture, on-chip memory type, processing power, and subsystems.

- TYPE A

  CPU + Flash Nonvolatile Memory with Digital and Analog Subsystems – Value Line G2xx Family
- TYPE B

  CPUXv2 + Flash Nonvolatile Memory with Digital and Analog Subsystems – F5xx Family
- TYPE C

  CPUXv2 + FRAM Nonvolatile Memory with Digital and Analog Subsystems – FR57xx Family

7.2.1 TYPE A: CPU + Flash Nonvolatile Memory with Digital and Analog Subsystems – Value Line G2xx Family

The MSP430 Value line G2xx MCUs are flash based devices that offer 16-bit MCU performance, up to 16-MHz operation, integrated intelligent peripherals, and industry leading ultra-low power. Flash and RAM memory technologies are used as on-chip nonvolatile and volatile memories, respectively. The architecture combined with five low-power modes, is optimized to achieve ultra low power operation. It has on-chip digitally controlled oscillator (DCO) to enable fast wake-ups from low-power modes. Integrated analog and digital peripherals include 10-bit ADC, analog comparator, 16-bit timers, GPIO, and serial communication modules that are device specific. Additionally, most MSP430 Value Line G2xx devices have a built-in pin oscillator function on some GPIO pins and this functionality may be used in capacitive touch sensing and other sensor applications to eliminate external passive components. Figure 5 shows the G2xx high-level block diagram with various functional safety features. These functional safety features have been categorized in functional safety features implemented in hardware (shown in the red callout blocks) and functional safety features enabled by MSP430 STL APIs (shown in the blue callout blocks).

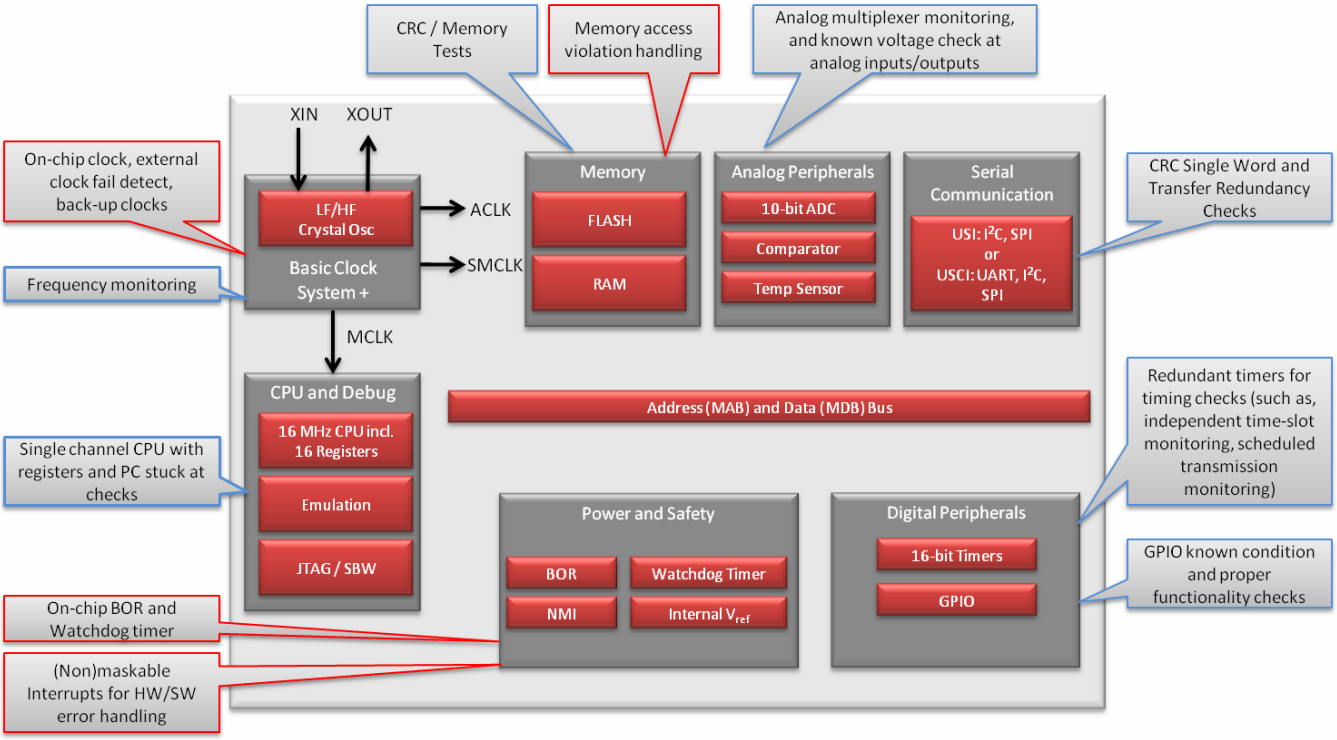

Figure 5. MSP430 Value Line G2xx High-Level Block Diagram With Hardware and Software Functional Safety Features

CPU

The 16-bit RISC CPU includes sixteen 16-bit registers (including program counter, stack pointer and status register) with full register access and single cycle register operations. The 16-bit address bus enables direct access and branching through the entire memory range, without the need for paging.

The STL includes software functions which allow checking CPU registers and the program counter (separately) for stuck at faults.

Exceptions and Error Handling

Hardware functional safety mechanism use interrupts and respective flags to indicate error occurrence and its cause to the CPU. Most of the critical fault indicating interrupts are categorized as system resets and (non)maskable interrupts and have higher interrupt priority compared to the peripheral interrupts. Also, these interrupts are not masked by the global interrupt enable (GIE) bit. Additionally, the $\overline{\text{RST}}$/NMI pin can be used to either reset the device from externally or trigger a (non)maskable interrupt to the CPU from external events when configured in reset and NMI modes, respectively.

Power Management

The always-on on-chip brown-out reset (BOR) circuitry detects low voltage condition on VCC (when supply voltage is either applied or removed to device's VCC terminal) and resets the device until supply voltage crosses the positive-going VCC threshold voltage.

Device Memories

The G2xx devices have nonvolatile flash and volatile RAM memories. Memory test functions such as software-based CRC checks and March tests are made available in the STL for MSP430 MCUs. These functions can be used to check for memory integrity either periodically (as periodic self test) or upon power-up (as power-up software routines). In device hardware, memory access violation detection is supported to indicate any incorrect or unpredictable memory accesses.

Clocks and Watchdog

On-chip clocks include a DCO and a VLO (very low power and low frequency oscillator). Some devices in the family have an integrated crystal oscillator that supports either low-frequency only mode or both low- and high-frequency modes. The STL provide frequency monitoring functionality used to track the main clock frequency variations (up to a threshold percentage value provided by user) by using a high-precision clock source (for example, an external LF crystal). In hardware, the oscillator fault fail-safe feature is supported to detect any oscillator faults and interrupt the CPU (if the interrupt enable bits are set) to take necessary actions. And, as part of fail-safe logic, if a fault is detected for the crystal oscillator that sources the main clock (MCLK), then the main clock source is automatically switched to the internal DCO.

The on-chip watchdog timer should be used to monitor critical software loops and operations and to perform a controlled system reset if a software problem arises and causes the watchdog timer to timeout.

Analog Peripherals

The analog peripherals on G2xx devices include ADC, internal voltage reference, comparator, and internal temperature sensor. The STL for MSP430 MCUs supports analog multiplexer monitoring functions to ensure proper operation of selectable analog channels, and plausibility check functions for ADCs and internal reference to ensure A/D results are within acceptable A/D drift count range. Similarly, proper comparator operation can be checked by comparing known external voltages against internal references.

Serial Communication

Integrated serial communication modules support industry standard interfaces such as: synchronous communication (SPI, I^2C) on devices with USI module, and both synchronous (SPI, I^2C) and asynchronous communication (UART) on devices with the USCI module. Software routines including transfer redundancy (where critical data is transmitted and receive multiple times) and CRC single word check (using CRC to monitor data integrity of message being transmitted or received in applications that are not time-constrained) can be implemented in user software to ensure reliable external communication. And, for devices with USCI peripheral, hardware features are supported like glitch suppression in UART mode and loopback mode in UART and SPI modes of operation.

Digital Peripherals

Digital peripherals include multiple 16-bit timers that can be used for supporting time base checks, and as redundant timers or channels to implement timing checks such as independent time-slot monitoring, scheduled transmission monitoring, and so on.

The GPIO pins on these devices are individually and independently programmable as inputs or outputs. They also support individual pullup or pulldown resistors and this feature can be used to prevent floating input condition of unused I/O pins. The STL for MSP430 MCUs support I/O functionality checks that can be performed either periodically or upon power-up. Also, based on the system application, user software can perform periodic known condition checks on GPIO port pins to monitor expected I/O states.

Other Safety

As an additional part of functional safety features, critical registers (such as flash control registers and WDT registers) are password protected to avoid inadvertent module configurations. This is especially important for safeguarding critical module operations when device is in an unknown operating state.

Appendix A shows functional safety diagnostics schemes that can be implemented to add diagnostic capability in each of these MCU functions during runtime and power-up.

7.2.2 TYPE B: CPUXv2 + Flash Nonvolatile Memory With Digital and Analog Subsystems – F5xx Family

The MSP430 F5xx family of devices features an extended 16-bit RISC CPUXv2 with up to 1MB memory access. This family of devices offer large memory footprints and has flash and SRAM memory technologies, which are used as on-chip nonvolatile and volatile memories, respectively. The architecture, combined with five low-power modes, is optimized to achieve ultra-low-power operation. It can operate up to 25 MHz and has on-chip digitally controlled oscillator (DCO) to enable fast wake-ups from low-power modes. Integrated peripherals include high precision analog modules such as 10-bit or 12-bit SAR ADC, analog comparator, and 12-bit DAC and advanced digital modules such as a hardware multiplier, 16-bit timers, an internal DMA, GPIOs, serial communication modules (which support I^2C, SPI, UART, IrDA), a real-time clock, and others. Additionally, some MSP430 F5xx devices have advanced peripherals such as integrated USB and PHY that support USB 2.0, high-resolution timer (Timer_D) that enables up to 4-ns resolution, encryption modules, and radio frequency (RF) connectivity. Figure 6 shows the F5xx high-level block diagram with various functional safety features. These functional safety features have been categorized in functional safety features implemented in hardware (shown in the red callout blocks) and functional safety features enabled by MSP430 STL APIs (shown in the blue callout blocks).

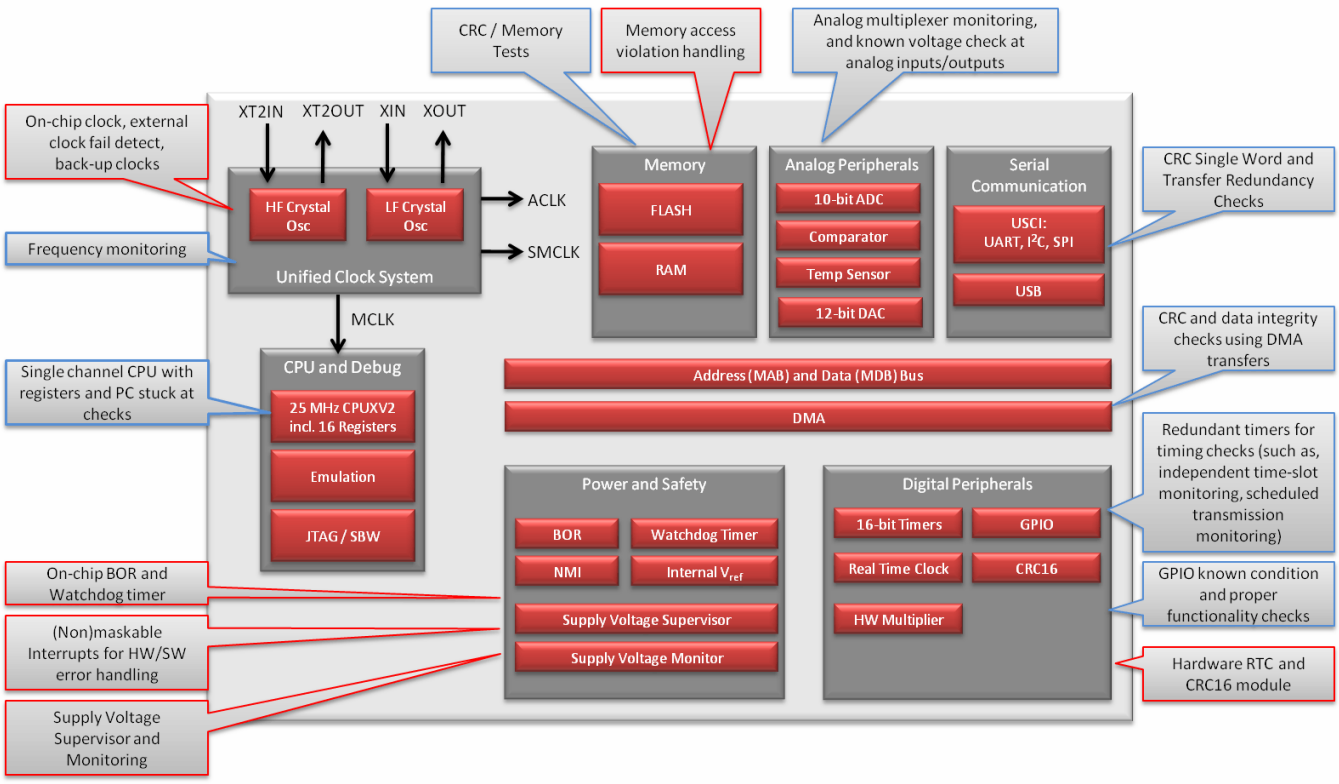

Figure 6. MSP430 F5xx High-Level Block Diagram with Hardware and Software Functional Safety Features

CPU

The extended 16-bit RISC CPUXv2 includes sixteen CPU registers (including program counter, stack pointer, and status register) with full register access and single cycle register operations. The 20-bit address bus enables direct access and branching through the entire memory range (up to 1MB), without the need for paging. The IEC 60730 software libraries for MSP430 MCUs have functions that check the CPU registers and the program counter (separately) for stuck at faults.

DMA

The MSP430 F5xx devices have an internal DMA with multiple channels to transfer data from one address to another without CPU intervention. Safety software routines required to access range of memory addresses (volatile, nonvolatile or peripheral memory) for implementing functions like CRC and other data integrity checks can use DMA for internal data transfer to ensure proper DMA operation. Currently, STL for MSP430 MCUs does not support using DMA transfers for nonvolatile and volatile memory checks. However, the STL takes advantage of device containing a CRC16 module for nonvolatile memory checks.

Exceptions and Error Handling

Hardware safety mechanism use interrupts and respective flags to indicate error occurrence and report the error cause back to the CPU. Most of the critical fault indicating interrupts are categorized as system resets and (non)maskable interrupts and have higher interrupt priority compared to the peripheral interrupts. Also, these interrupts are not masked by the global interrupt enable (GIE) bit. Additionally, the $\overline{\text{RST}}$/NMI pin can be used to either reset the device from externally or trigger a (non)maskable interrupt to the CPU from external events when configured in reset and NMI modes, respectively.

Power Management

The always-on on-chip brown-out reset (BOR) circuitry detects low voltage condition on VCC (when supply voltage is either applied to or removed from the device's VCC terminal) and resets the device until supply voltage crosses the positive-going VCC threshold voltage. Additionally, the MSP430 F5xx devices have an advanced power management module (PMM) with integrated supply voltage supervisor (SVS) and supply voltage monitor (SVM) circuitry with user-selectable thresholds. Depending on the system application, the SVS and SVM circuits can be used to reset the device or interrupt the CPU when supply voltage drops below the user-selected threshold voltages.

Device Memories

The F5xx devices have nonvolatile flash and volatile RAM memories. Memory test functions such as hardware-based CRC checks (for devices which contain CRC16 module) or software-base CRC checks (for device without CRC16 module) and March tests are made available in the STL for MSP430 MCUs. These functions can be used to check for memory integrity either periodically (as periodic self test) or upon power-up (as power-up software routines). Additionally, the F5xx devices have a hardware CRC module, which can be used for 16-bit CRC computations. Also, detection of memory access violations is supported in device hardware to indicate any incorrect or unpredictable memory accesses.

Clocks

On-chip clocks include the DCO that can be optionally stabilized by an frequency-locked loop (FLL), REFO (which is a trimmed low-frequency oscillator with a 32768-Hz typical frequency), and a VLO (very low-power and low-frequency oscillator). These devices also have integrated crystal oscillators (XT1 supports either only low-frequency mode or both low- and high-frequency modes, and XT2 supports only high-frequency mode). The STL for MSP430 MCUs provide frequency monitoring functionality used to track the main clock frequency variations (up to a threshold percentage value provided by user) by using a high-precision clock source (for example, external LF crystal). In hardware, fail-safe features such as oscillator fault fail-safe and DCO fault flag features are supported to detect any oscillator or DCO faults. Upon detecting a fault condition, the CPU can be interrupted (by setting respective interrupt enable bits) to take necessary actions. And, as part of fail-safe logic, if a fault is detected for crystal oscillators, then the respective clock sources are automatically switched to internal clocks (either DCO or REFO, depending on the oscillator that failed and the system clocks they were sourcing). Refer to the *MSP430x5xx and MSP430x6xx Family User's Guide* (SLAU208) "Unified Clock System" section for more details.

The on-chip watchdog timer should be used to monitor critical software loops and operations, and to perform a controlled system reset if a software problem arises and causes the watchdog timer to timeout.

Analog Peripherals

The analog peripherals on F5xx devices include 10-bit or 12-bit ADC, internal voltage reference, internal temperature sensor, comparator and 12-bit DAC. The STL for MSP430 MCUs support analog multiplexer monitoring functions to ensure proper operation of selectable analog channels, and plausibility check functions for ADCs and internal reference to ensure A/D results are within acceptable A/D drift count range. Similarly, proper comparator operation can be checked by comparing known external voltages against internal references. As part of plausibility check for the DAC, particular digital codes can be forced as D/A inputs and unused A/D input channels can be used to check for expected D/A outputs.

Serial Communication

Integrated serial communication modules, like USCI, support industry standard interfaces such as UART, SPI and I^2C. Some F5xx devices have an integrated USB module that supports the USB 2.0 standard, which inherently ensures data integrity. Software routines including transfer redundancy (where critical data is transmitted and receive multiple times) and CRC single word check (using CRC to monitor data integrity of message being transmitted or received in applications that are not time-constrained) can be implemented in user-software to ensure reliable external communication. Also, hardware features are supported like glitch suppression in UART mode and loopback mode in UART and SPI modes of operation.

Digital Peripherals

Digital peripherals include multiple 16-bit timers that can be used for supporting time base checks, and redundant timers or channels can be used to implement timing checks, such as, independent time-slot monitoring, scheduled transmission monitoring and so on. Additionally, the F5xx devices have an integrated real-time clock (RTC) module that can be used for real time monitoring based safety checks in user application.

The GPIO pins on these devices are individually and independently programmable as inputs or outputs. They also support individual pullup or pulldown resistors, and this feature can be used to prevent floating input condition of unused I/O pins. The STL for MSP430 MCUs support I/O functionality checks that can be performed either periodically or upon power-up. Also, based on the system application, user software can perform periodic known condition checks on GPIO port pins to monitor expected I/O states.

Other Functional Safety Considerations

Additionally, as a part of hardware functional safety, critical registers (such as flash control registers and WDT registers) are password protected to avoid inadvertent module configurations. This is especially important for safeguarding critical module operations during incorrect device operation.

Appendix A shows many of the functional safety diagnostics schemes that can be implemented to add diagnostic capability in each of these MCU functions during runtime and power-up.

7.2.3 TYPE C: CPUXv2 + FRAM Nonvolatile Memory With Digital and Analog Subsystems – FR57xx Family

The MSP430 FR57xx family of devices features embedded FRAM nonvolatile memory, extended 16-bit MSP430 CPU (CPUXv2), and different peripherals targeted for various applications. The architecture, FRAM, and peripherals, combined with seven low-power modes, are optimized to achieve ultra-low-power operation. FRAM is a nonvolatile memory that combines the speed, flexibility, and endurance of SRAM with the stability and reliability of flash, all at lower total power consumption. These devices can operate up to 24 MHz and have an on-chip digitally controlled oscillator (DCO) to enable fast wake-ups from low-power modes. Integrated analog and digital peripherals include a 10-bit ADC, a 16-channel analog comparator, enhanced serial communication modules that support I^2C, SPI, or UART protocols, an internal DMA, a hardware multiplier, a real-time clock, 16-bit timers, and more. Figure 7 shows the FR57xx high-level block diagram with various functional safety features. These functional safety features have been categorized in functional safety features implemented in hardware (shown in the red callout blocks) and functional safety features enabled by MSP430 STL APIs (shown in the blue callout blocks).

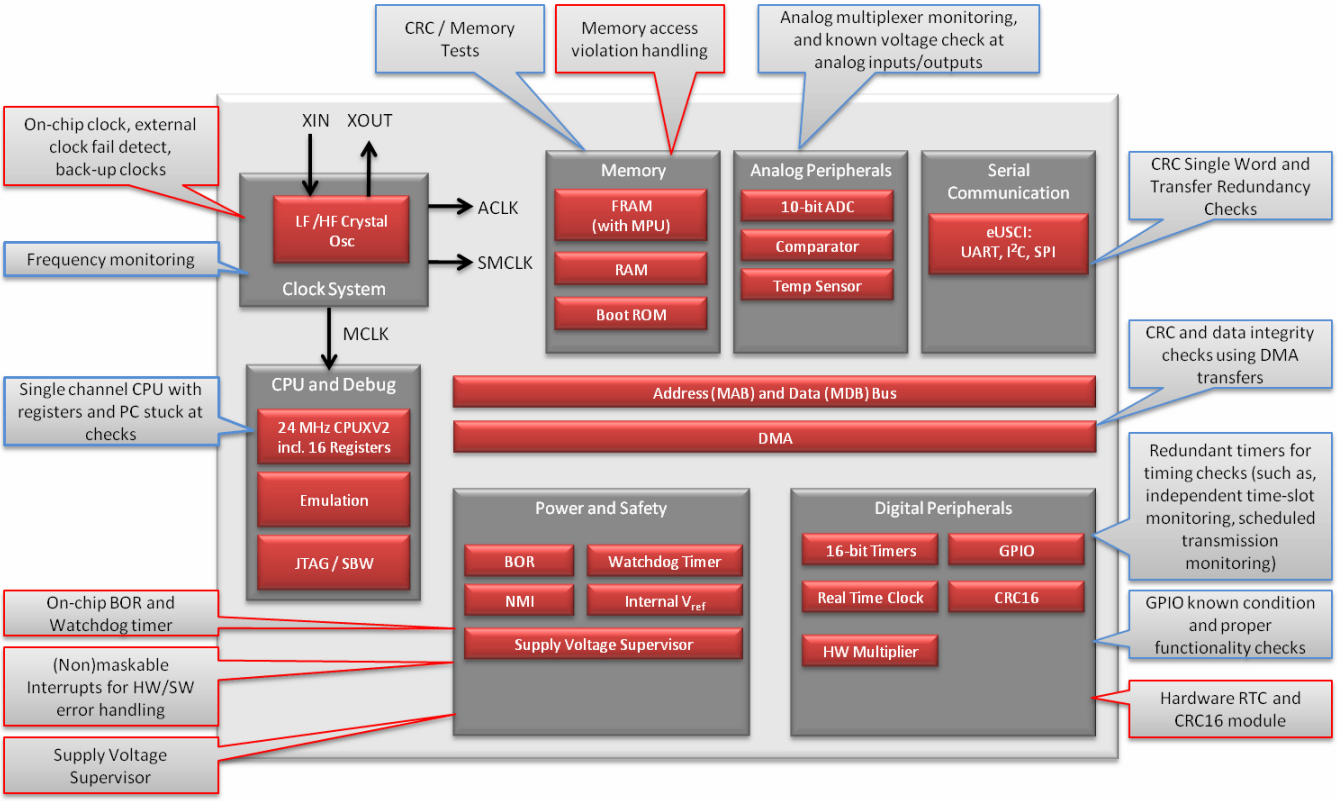

Figure 7. MSP430 FR57xx High-Level Block Diagram with Hardware and Software Functional Safety Features

CPU

The extended 16-bit RISC CPUXv2 includes sixteen CPU registers (including program counter, stack pointer and status register) with full register access and single cycle register operations. The 20-bit address bus enables direct access and branching through the entire memory range (up to 1MB), without the need for paging. The MSP430 STL have functions that check the CPU registers and the program counter (separately) for stuck at faults.

DMA

The MSP430 FR57xx devices have an internal DMA with multiple channels to transfer data from one address to another without CPU intervention. Safety software routines required to access range of memory addresses (volatile, nonvolatile or peripheral memory) for implementing functions like CRC and other data integrity checks can use DMA for internal data transfer to ensure proper DMA operation. Currently, STL for MSP430 MCUs does not support using DMA transfers for nonvolatile and volatile memory checks. However, the STL takes advantage of device containing a CRC16 module for nonvolatile memory checks.

Exceptions and Error Handling

Hardware safety mechanism use interrupts and respective flags to indicate error occurrence report the error cause back to the CPU. Most of the critical fault indicating interrupts are categorized as system resets and (non)maskable interrupts and have higher interrupt priority compared to the peripheral interrupts. Also, these interrupts are not masked by the global interrupt enable (GIE) bit. Additionally, the $\overline{\text{RST}}$/NMI pin can be used to either reset the device from externally or trigger a (non)maskable interrupt to the CPU from external events when configured in reset and NMI modes, respectively.

Power Supply and Management

The always-on on-chip brown-out reset (BOR) circuitry detects low voltage condition on VCC (when supply voltage is either applied or removed to device VCC) and resets the device until supply voltage crosses the positive-going VCC threshold voltage. Additionally, the MSP430 FR57xx devices have an advanced power management module (PMM) with integrated supply voltage supervisor (SVS) circuitry with fixed voltage thresholds.

Device Memories

The FR57xx devices have embedded FRAM nonvolatile memory and volatile RAM memory. FRAM offers unified memory with dynamic partitioning and an integrated memory protection unit (MPU) in hardware is used to protect against inadvertent writes to designated read-only memory segments or inadvertent code execution from constant memory segment. Memory test functions such as hardware-based CRC checks (for devices that contain CRC16 module) or software-base CRC checks (for device without a CRC16 module) and March tests are made available in the STL for MSP430 MCUs. These functions can be used to check for memory integrity either periodically (as periodic self test) or upon power-up (as power-up software routines). Additionally, the FR57xx devices have a hardware CRC module, which can be used for 16-bit CRC computations. Also, detection of memory access violations is supported in device hardware to indicate any incorrect or unpredictable memory accesses.

Clocks

On-chip clocks include DCO with three selectable fixed frequencies (up to 24MHz), and a VLO (very low-power and low-frequency oscillator). These devices also have integrated crystal oscillators (XT1 supports either only low-frequency mode or both low- and high-frequency modes, and XT2 supports only high-frequency mode). The IEC 60730 software libraries for MSP430 MCUs provide frequency monitoring function used to track the main clock frequency variations (up to a threshold percentage value provided by user) by using a high-precision clock source (for example, an external LF crystal). In hardware, fail-safe features such as oscillator fault fail-safe is supported to detect any oscillator faults. Upon detecting a fault condition, the CPU can be interrupted (by setting interrupt enable bits) to take necessary actions. And, as part of fail-safe logic, if a fault is detected for crystal oscillators, then the respective clock sources are automatically switched to internal clocks (either DCO or VLO, depending on the oscillator that failed and the system clocks they were sourcing). Refer to the *MSP430FR57xx Family User's Guide* (SLAU272) "Clock System" section for more details.

The on-chip watchdog timer should be used to monitor critical software loops and operations, and to perform a controlled system reset if a software problem arises and causes the watchdog timer to timeout.

Analog Peripherals

The analog peripherals on FR57xx devices include 10-bit ADC, internal voltage reference, internal temperature sensor, and comparator. The IEC 60730 software libraries for MSP430 MCUs support analog multiplexer monitoring functions to ensure proper operation of selectable analog channels, and plausibility check functions for ADCs and internal reference to ensure A/D results are within acceptable A/D drift count range. Similarly, proper comparator operation can be checked by comparing known external voltages against internal references.

Serial Communication

Integrated serial communication modules, like enhanced USCI (eUSCI), support industry standard interfaces such as UART, SPI, and I^2C. Software routines including transfer redundancy (where critical data is transmitted and receive multiple times) and CRC single word check (using CRC to monitor data integrity of message being transmitted or received in applications that are not time-constrained) can be implemented in user-software to ensure reliable external communication. And, hardware features are supported like glitch suppression with configurable deglitch time in UART and I^2C modes and loopback mode in UART and SPI modes of operation.

Digital Peripherals

Digital peripherals include multiple 16-bit timers that can be used for supporting time base checks, and redundant timers/channels can be used to implement timing checks, such as, independent time-slot monitoring, scheduled transmission monitoring and so on. Additionally, the MSP430 FR57xx devices have an integrated real-time clock (RTC) module that can be used for real time monitoring based safety checks in user application.

The GPIO pins on these devices are individually and independently programmable as inputs or outputs. They also support individual pullup or pulldown resistors and this feature can be used to prevent floating input condition of unused I/O pins. The IEC 60730 software libraries for MSP430 MCUs support I/O functionality checks that can be performed either periodically or upon power-up. Also, based on the system application, user software can perform periodic known condition checks on GPIO port pins to monitor expected I/O states.

Other Functional Safety Considerations

Additionally as a part of hardware safety, critical registers (such as FRAM and MPU control registers and WDT registers) are password protected to avoid inadvertent module configurations. This is especially important for safeguarding critical module operations during incorrect device operation.

Appendix A shows many of the functional safety diagnostics schemes that can be implemented to add diagnostic capability in each of these MCU functions during runtime and power-up.

7.3 Management of Family Variants Supported by MSP430 STL

The MSP430 Value Line G2xx, F5xx, and FR57xx families all include multiple product variants. Table 2 shows an overview of the CPU, memories, and peripherals that each of the families support. In each of these families, there are a number of devices each with different combinations of peripherals. To find the CPU, memory, and peripheral mix for any given device in each of these families, see device-specific data sheet.

These devices are implemented as either unique silicon designs or as shared silicon designs that have elements (peripherals or features) disabled or not specified, even if present on silicon. Only the elements that are enabled in the device and specified in the device-specific data sheet are to be used for safety feature enhancements or safety software implementation. Elements that are not part of the device, even if they are supported in the superset of the device family, are not ensured to be present or to operate.

Table 2. MSP430 G2xx, F5xx, and FR57xx Features List(1)

	MSP430 16-Bit Microcontrollers		
	G2xx	F5xx	FR57xx
16-bit RISC CPU	16 MHz	25 MHz	24 MHz
Extended 16-bit CPU (CPUXv2)(2)		X	X
Extended 16-bit CPU (CPUX)	X		
Constant Generator	X	X	X
DMA		X	X
Emulation Logic	X	X	X
Memory			
Boot ROM			X
RAM	X	X	X
Flash	X	X	
FRAM			X
Digital Peripherals			
16-bit Timers (Timer_A, Timer_B)	X(3)	X	X
High-Resolution Timer (Timer_D)		X(3)	
Real-Time Clock		X	X
Watchdog Timer	X	X	X
CRC Module		X	X
16-bit/32-bit Hardware Multiplier		X	X
AES Accelerator		X	

(1) Module or peripheral version is dependent on the device family within which it is available. For example, the comparator module in Value Line G2xx devices is Comp_A+, in F5xx devices it is Comp_B, and in FR57xx devices it is Comp_D. Refer to the family-specific user's guide for details.
(2) CPUXv2 supports up to 1MB memory access through a 20-bit address bus and CPU registers.
(3) Module or peripheral availability within the device family is specific to the device variant. Refer to the device-specific data sheet for these details.

Table 2. MSP430 G2xx, F5xx, and FR57xx Features List[1] (continued)

	MSP430 16-Bit Microcontrollers		
	G2xx	F5xx	FR57xx
Communication Modules			
USI (I^2C or SPI)	X[3]		
USCI (I^2C, SPI, or UART)	X[3]	X	
Enhanced USCI (eUSCI) (I^2C, SPI, or UART)			X
USB		X[3]	
Analog Peripherals			
10-bit ADC	X[3]	X[3]	X[3]
12-bit ADC		X[3]	
12-bit DAC		X[3]	
Comparator	X[3]	X[3]	X
Temperature Sensor	X[3]	X[3]	X
Voltage Reference	X[3]	X	X
LF, HF Crystal Oscillator	X[3]	X[3]	X
BOR Circuitry[4]	X	X	X
Supply Voltage Supervisor		X	X
Supply Voltage Monitor		X	

(4) In the F5xx and FR57xx families, the BOR circuitry triggers a BOR reset. In the G2xx family, the BOR circuitry triggers a POR reset.

7.4 System Reset and Initialization

The system reset circuitry handles all device reset and initialization events. This circuitry is device family specific and comprises the following reset triggers:

- Brownout reset (BOR) (Not applicable to the MSP430 Value Line G2xx family—in the G2xx family, the BOR circuitry triggers a POR reset.)
- Power on reset (POR)
- Power up clear (PUC)

The reset states mentioned above are triggered by specific events that are also device family specific. Table 3 summarizes the list of events that trigger the various reset states in each device family.

Table 3. MSP430 System Reset Triggers[1][2]

Reset State	G2xx	F5xx	FR57xx
BOR	Not Applicable	• Device power-up • $\overline{\text{RST}}$/NMI pin • LPMx.5 wake-up • Software BOR	• Device power-up • $\overline{\text{RST}}$/NMI pin • LPMx.5 wake-up • Supply Voltage Supervisor condition • Software BOR
POR	• Device power-up • $\overline{\text{RST}}$/NMI pin	• BOR signal • Supply Voltage Supervisor condition • Software POR trigger	• BOR signal • Software POR trigger
PUC	• POR signal • Watchdog events • Password violation events • Fetch from peripheral area	• POR signal • Watchdog events • Password violation events • Fetch from peripheral area	• POR signal • Watchdog events • Password violation events • Fetch from peripheral area

(1) For more details about family-specific reset states triggers, refer to the family-specific user's guide in the "System Resets, Interrupts, and Operating Modes" section.

(2) The number and types of reset triggers that are available may vary from device to device within the same family. See the device-specific data sheet for all reset sources that are available.

7.5 Device Operation After System Reset

After a system reset condition, the device enters active mode, and the program counter (PC) is loaded with user application start address contained at the reset vector location at 0xFFFEh. In devices where system reset events trigger a BOR, the bootcode is first executed to load the factory stored calibration values of the on-chip oscillator and reference voltages, and then the PC is loaded with the address at reset vector location.

The user software should initialize the device for the application requirements. This includes initializing stack pointer, initializing watchdog timer, and configuring the peripheral modules that are used in the application. For details regarding the peripherals and necessary configuration, refer to the family-specific user's guide (see Table 4).

7.6 Device Operation in Low-Power Modes

The MSP430 device low-power modes are designed considering ultra-low power, speed, and data throughput and current consumption of individual peripherals.

The device is in active mode when the CPU is active and executing user-application code. When the user application does not require the CPU to be active and is instead waiting for interrupts from various peripherals to continue operation, then, depending on the peripherals that are operational in the background (without CPU operation) and the clock and power sources they are requesting, the device can be placed in different low-power modes to reduce the overall power consumption.

For more details regarding the various low-power modes supported, the respective bit definitions and clock availability in different modes refer to the device family user's guide as listed in Table 4.

7.7 Management of Exception and Errors

The MSP430 MCU device architectures leverage CPU interrupts for event triggers. All hardware safety mechanism use these interrupt flags to indicate error occurrence and its cause to the CPU. There are three types of CPU interrupts:

- System Reset
- (Non)maskable
- Maskable

The number and types of interrupts under each category vary from device to device. Upon system reset, respective system reset interrupt flags are set indicating the cause of device reset. All maskable and (non)maskable interrupts can be individually masked by interrupt enable bits, and additionally, all maskable interrupts can be disabled by a general interrupt enable (GIE) bit.

Interrupt priorities determine what interrupt is serviced when more than one interrupt is pending simultaneously. The interrupt priorities on MSP430 devices are fixed. System reset interrupts have the highest priority, followed by (non)maskable interrupts, and then maskable interrupts. Maskable interrupts are used by peripherals with interrupt capability and the priority definition of these interrupts varies from device to device. See device-specific data sheet for these details.

The IEC 60730 library example projects support an error and exception reporting function using two General Purpose Input and Output (GPIO) which allows system monitoring during validation and runtime use of the application. This function can be easily customized in the end application based on the error reporting needs.

7.8 MSP430 General Safety Mechanism and Assumptions of Use

7.8.1 Power Supply and Power Management Module (PMM)

The MSP430 MCUs require an external supply or voltage regulator to provide necessary voltages and currents for proper device operation. For devices with separate analog and digital power pins (AVCC and DVCC), it is recommended to power them both from the same source as the MSP430 MCUs can tolerate a maximum difference of 0.3 V between AVCC and DVCC pins during power-up and device operation. The MSP430 F5xx and FR57xx devices have an integrated LDO to produce a secondary core voltage from the primary supply (applied at DVCC) which is used to power CPU, memories and digital modules and this greatly enhances the power efficiency of the system.

The MSP430 G2xx and F5xx devices support a wide supply voltage range of 1.8 V to 3.6 V. The FR57xx devices support supply voltages ranging from 2.0 V to 3.6 V. All MSP430 MCUs have an always-on on-chip BOR circuitry to detect supply under-voltage conditions during power-up and power-down scenarios and to hold the device in reset state until the supply voltage crosses the BOR positive-going threshold voltage.

Additionally, the MSP430 F5xx family has an advanced power management module (PMM) with supply voltage supervisor (SVS) and supply voltage monitor (SVM) circuits. Both the SVS and SVM circuits feature software selectable thresholds. Depending on the system requirement, the user-application can configure the SVS and SVM with respective thresholds and select either a device reset trigger or an interrupt to the CPU when the supply voltage drops below the software-selected threshold values. Also, the SVM circuitry can be used for over-voltage detection to detect rising DVCC that exceeds safe device operation.

The MSP430FR57xx family of devices features a PMM with a supply voltage supervisor (SVS) with fixed thresholds to handle power-up and power-down scenarios.

External voltage supervisors for under-voltage and over-voltage conditions are necessary if the on-chip detection thresholds are not meeting the performance goals. It is the responsibility of the system integrator to make the safety tradeoffs.

The MSP430 G2xx family of devices requires external supervisors or monitors, as they do not have internal supply voltage supervisor or monitor circuits. MSP430G2xx family have only an integrated BOR circuit.

7.8.2 External Voltage Supervisor

For devices with no overvoltage supervisor or monitor circuitry, it is highly recommended to use an external voltage supervisor to monitor for an overvoltage condition on the input supply voltage. Overvoltage threats are more common in industrial products. However, it is especially important to include a external voltage supervisor if it is required to avoid catastrophic shutdown of the system upon detection of supply overvoltage condition.

7.8.3 Reset

The system reset circuitry in MSP430 MCUs helps reset the device to place all the synchronous and asynchronous device logic into a known state. Depending on the device family, different reset states (BOR, POR and PUC), and various events that trigger these reset states are supported. The reset can be triggered externally (for example, by power-up, $\overline{\text{RST}}$/NMI pin, or LPMx.5 wakeup) or by the device itself as a part of system or application fault indication (for example, a watchdog event, password violation, supply voltage supervisor and monitor condition, or software reset). Refer to the family-specific user's guide and device-specific data sheet for more details about reset states and reset triggers.

To avoid inadvertent device resets due to external noise on the reset pin, it is recommended to terminate the reset pin with a 47-kΩ pullup resistor and 10-nF pulldown capacitor. Note that the pulldown capacitor should not exceed 2.2 nF when using the Spy-Bi-Wire debug interface.

7.8.4 Clocks

The MSP430 MCUs support a variety of on-chip clocks (DCO to support high system frequencies, internal 32-kHz REFO, and very low-power and low-frequency VLO oscillator) and integrated crystal oscillators (XT1 that supports either low-frequency mode only or both low- and high-frequency modes, and XT2 that supports high-frequency mode). Also, the clock systems support oscillator fault indication and fail safe features. The on-chip clock architecture, availability and fault-fail-safe features are device family dependent. Section 7.2 gives a high-level overview of the clock systems in MSP430 G2xx, F5xx, and FR57xx devices. For more details regarding device clocks, refer to the family-specific user's guide (see Table 4).

7.9 Recommended MSP430 Documentation

This *Safety Manual for MSP430G2xx, MSP430F5xx, and MSP430FR57xx Devices in IEC 60730 Safety Applications* is complemented by the device documentation that is shown in Table 4. This documentation provides the necessary details on functional implementation and application use.

Table 4. MSP430 MCU Documentation

MSP430 MCU Product Family Technical Documentation	Abstract
MSP430G2xx Family User's Guide (SLAU144) MSP430F5xx Family User's Guide (SLAU208) MSP430FR57xx Family User's Guide (SLAU272)	Reference manual that describes the modules and peripherals of each family of devices(1)
Device-specific data sheet	Provides device-specific pin functions, internal signal connections, and operational parameters
Device-specific errata sheet	Provides a list of errata that affect specific silicon revisions of devices

(1) Not all features and functions of all modules or peripherals are present on all devices. In addition, modules or peripherals may not be fully implemented on an individual device in the same family. Pin functions, internal signal connections, and operational parameters differ from device to device. The user should refer to device-specific data sheet for these details.

8 Next Steps in Your Safety Development

TI's support for your safety development does not stop with the delivery of this safety manual. You have a wide range of support options that includes:

- SafeTI™ IEC 60730 design packages and product support on the web: http://www.ti.com/ww/en/functional_safety/safeti/SafeTI-60730.html
- Access MSP430 MCU page on the web: http://www.ti.com/lsds/ti/microcontroller/16-bit_msp430/overview.page
- The MSP430 wiki page provides answers to many commonly asked questions: http://processors.wiki.ti.com/index.php/MSP430_FAQ
- MSP430 wiki page provides best practices using flash memory on MSP430 devices: http://processors.wiki.ti.com/index.php/MSP430_Flash_Best_Practices

Software Best Practices for Functional Safety Applications

The MSP430 MCUs have several functional safety features that can initiate fault or error interrupts when a fault is detected. However, except for a few hardware modules, not all of these modules are able to generate alarms in their dormant state. The MSP430 STL helps to monitor all of the MCU regions that are required for IEC 60730 Class B compliance, which includes the CPU, memory, and a few peripherals, and to generate error conditions if they show any abnormalities. It is highly recommended that in addition to POST and PEST execution in the application firmware, developers implement the following software functionality in their applications.

Table 5 shows commonly used safety diagnostic software functions that enable safety features and attributes across subsystems. These functions are the recommended common methods that can be enhanced as needed for each application.

Table 5. Software Best Practices for Safety Applications in MSP430 MCUs

Safety Diagnostics Functions to Enable Safety Attributes and Features	Description
Self Test	Periodic execution enables inherent self-checking features. This helps to detect latent faults.
Periodic Read Back	Periodic read back of configuration registers can provide a diagnostic for inadvertent writes or disturb of these registers. Error response, diagnostic testability, and any necessary software requirements are defined by the software implemented by the system integrator.
Read Back of Written Configuration	In order to ensure proper configuration of memory-mapped control registers, it is highly recommended that software implements a test to confirm proper operation of all control registers
Interrupt Sweep	MSP430 interrupt priorities are fixed and defined by the arrangement of the modules in the connection chain as shown in the interrupts section of device-specific data sheet. Interrupt sweep tests are internal self-tests that trigger interrupts when the CPU sets the interrupt flag registers. This function enables all the interrupt flags, sequentially, blocking the peripheral sources to check periodicity and correct occurrence of interrupts.
Run Time Code Trace	This function is a practical and easy method to trace the program execution to detect orderly execution of the critical interrupts among the configured peripherals. In most systems, program code jumps and branches in a predetermined code execution path, particularly in critical control loops. Tracing the hardware or software semaphores as a signature of orderly entry and exit checks proper timing of interrupts and peripheral functionality.
CRC and Read and Write	The MSP430G2xx and MSP430F5xx families do not support ECC and parity logic for memory blocks to provide error correction. To enable the memory safety, a CRC calculation and periodic memory read and write operation must be implemented. The MSP430 FR57xx family supports ECC with bit error correction capabilities, extended error detection and flag indicator. However, at an application level a CRC calculation and periodic memory read and write operation is recommended.
Loopback Tests	Not all communication ports have data integrity check during transmit sessions. However, most of these peripherals (USCI and eUSCI) have an internal loopback. This function is intended to periodically enable loopback mechanism and perform self diagnostic on the peripherals. It is recommended these tests are done with IO mux in input mode to avoid external signal visibility at the pin level, during test.
Custom Application Module	Most MCU firmware implements custom hardware or software functions intellectual property (IP) to differentiate the application. These are either ROM and flash-/FRAM-based or RAM-based functions that are invoked during runtime of the application. These functions should have explicit data integrity checks (program code CRC) and functional checks using test vectors to make sure that the custom software block functions correctly for the intended application.

Table 5. Software Best Practices for Safety Applications in MSP430 MCUs (continued)

Safety Diagnostics Functions to Enable Safety Attributes and Features	Description
1oo2 Software Voting	One out of two (1oo2) channels scheme is a method to implement software voting of correct functioning using similar peripherals, with one of the spare peripherals acting as a reference to compare. For example, use spare and redundant Timer resources to compare the correct functioning of the primary Timer resources.
Information Redundancy Techniques	Information redundancy techniques can be applied via software as an additional runtime diagnostic on any memory block, serial communication peripherals and control peripherals. There are many techniques that can be applied, such as read back of written values and multiple reads of the same target data with comparison of results. Error response, diagnostic testability, and any necessary software requirements are defined by the software implemented by the system integrator. The use of information redundancy techniques as a means to do functional safety diagnostic is a highly recommended or recommended function which depends in the criticality of the peripheral block in the end system.
Stack monitoring	During application reset, the application can analyze the stack of the _previous_ session to see how many bytes were left free and log for diagnostic purposes during application runtime. Periodically check the contents of the stack pointer. This can be checked within an interrupt service routine (ISR). Warn or reset the application if a certain critical threshold is reached.

MSP430 MCU Development Process for Management of Systematic Faults

For a safety critical development, it is necessary to manage both systematic and random faults. Texas Instruments have created a unique development process for safety critical semiconductors that greatly reduces probability of systematic failure. This process builds on a standard quality managed development as the foundation for safety critical development. The TI MCU teams use this process across all its MSP430 family of MCUs.

B.1 MSP430 MCU Development Process

All new product development at TI follows a structured new product development process. A phase review system for each group is defined in controlled documented procedures. A formal project review and approval by responsible management is completed and documented at critical points in the development process. The process is designed to manage organizational interfaces, project risks, and communication between groups involved in the development process.

Though not explicitly developed for compliance to a functional safety standard, the TI standard MCU development process already features many elements necessary to manage systematic faults. This development process can be considered to be Quality Managed (QM), but does not support achievement of an IEC 61058 Safety Integrity Level (SIL) or ISO 26262 Automotive Safety Integrity Level (ASIL). The TI standard MCU development process is certified to be compliant to ISO TS 16949:2009 as assessed by Bureau Veritas Certification under IATF Certificate No. 148738 Rev 1, Buerau veritas certification No. USA-13036/9-TS dated 20 October 2012.

The standard development process breaks development into phases:

- Business opportunity pre-screen
- Program planning
- Create
- Validate, sample and characterize
- Qualify
- Ramp to production and sustaining production

Figure 8 shows the Texas Instruments Incorporated (TI) standard silicon development process.

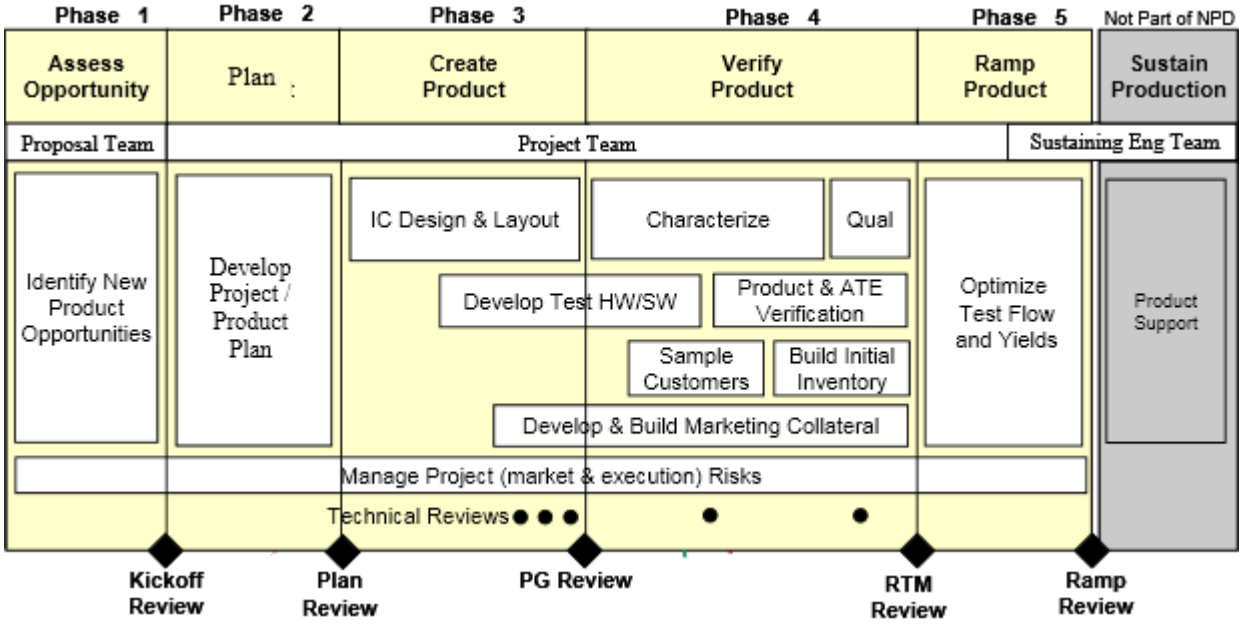

Figure 8. TI Standard MCU QM Development Process

Glossary

Table 6. Glossary

Terminology and Abbreviations	Definition and Full Expression
Class B	Class B is IEC 60730 classification of safety standards for appliances that use a microelectronics controller (MCU).
FRAM	Ferroelectric random access memory
POST	Power-on self test
PEST	Periodic self test
STL	Self-test library
ISR	Interrupt service routine

Revision History

Changes from January 11, 2014 to January 13, 2016 **Page**

- Added note to the "BOR Circuitry" entry in Table 2, *MSP430 G2xx, F5xx, and FR57xx Features List* 22
- Added to description in "Brownout reset (BOR)" list item in Section 7.4, *System Reset and Initialization* 22

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

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

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