Functional Safety Information

Functional Safety Manual for MSPM0L130x-Q1

TEXAS INSTRUMENTS

Table of Contents

List of Figures

List of Tables

1 Introduction

This document is a functional safety manual for the Texas Instrument MSPM0L component. The part numbers supported by this functional safety manual are as follows:

- MSPM0L130x-Q1

This functional safety manual provides information needed by system developers to help in the creation of a functional safety system using a MSPM0L component. This document includes:

- An overview of the component architecture
- An overview of the development process used to decrease the probability of systematic failures
- An overview of the functional safety architecture for management of random failures
- The details of architecture partitions and implemented functional safety mechanisms

The following information is documented in the *[Functional Safety Analysis Report]* and is not repeated in this document:

- Summary of failure rates (FIT) of the component
- Summary of functional safety metrics of the hardware component for targeted standards (for example IEC 61508, ISO 26262, and so forth)
- Quantitative functional safety analysis (also known as FMEDA, Failure Modes, Effects, and Diagnostics Analysis) with detail of the different parts of the component, allowing for customized application of functional safety mechanisms
- Assumptions used in the calculation of functional safety metrics

The following information is documented in the *[Functional Safety Report]*, and is not repeated in this document:

- Results of assessments of compliance to targeted standards

The user of this document should have a general familiarity with the MSPM0L component. For more information, refer to the data sheet. This document is intended to be used in conjunction with the pertinent data sheets, technical reference manuals, and other component documentation.

For information that is beyond the scope of the listed deliverables, contact your TI sales representative or go to Function safety.

Trademarks

TI E2E™ is a trademark of Texas Instruments.
All trademarks are the property of their respective owners.

2 MSPM0L Hardware Component Functional Safety Capability

This section summarizes the component functional safety capability.

This hardware component:

- Was developed as a functional Safety Element out of Context (SEooC)
- Was developed according to the relevant requirements of ISO 26262:2018
- Aids in achieving systematic integrity up to ASIL-B
- Includes functional safety mechanisms for random fault integrity requirements to aid in achieving systematic integrity up to ASIL-B

3 Development Process for Management of Systematic Faults

For functional safety development, it is necessary to manage both systematic and random faults. Texas Instruments follows a new-product development process for all of its components which helps to decrease the probability of systematic failures. This new-product development process is described in Section 3.1. Components being designed for functional safety applications will additionally follow the requirements of TI's functional safety development process, which is described in Section 3.2.

3.1 TI New-Product Development Process

Texas Instruments has been developing components for automotive and industrial markets since 1996. Automotive markets have strong requirements regarding quality management and product reliability. The TI new-product development process features many elements necessary to manage systematic faults. Additionally, the documentation and reports for these components can be used to assist with compliance to a wide range of standards for customer's end applications including automotive and industrial systems (for example, ISO 26262-4, IEC 61508-2).

This component was developed using TI's new product development process which has been certified as compliant to ISO 9001 / IATF 16949 as assessed by Bureau Veritas (BV).

The standard development process breaks development into phases:

- Assess
- Plan
- Create
- Validate

Figure 3-1 shows the standard process.

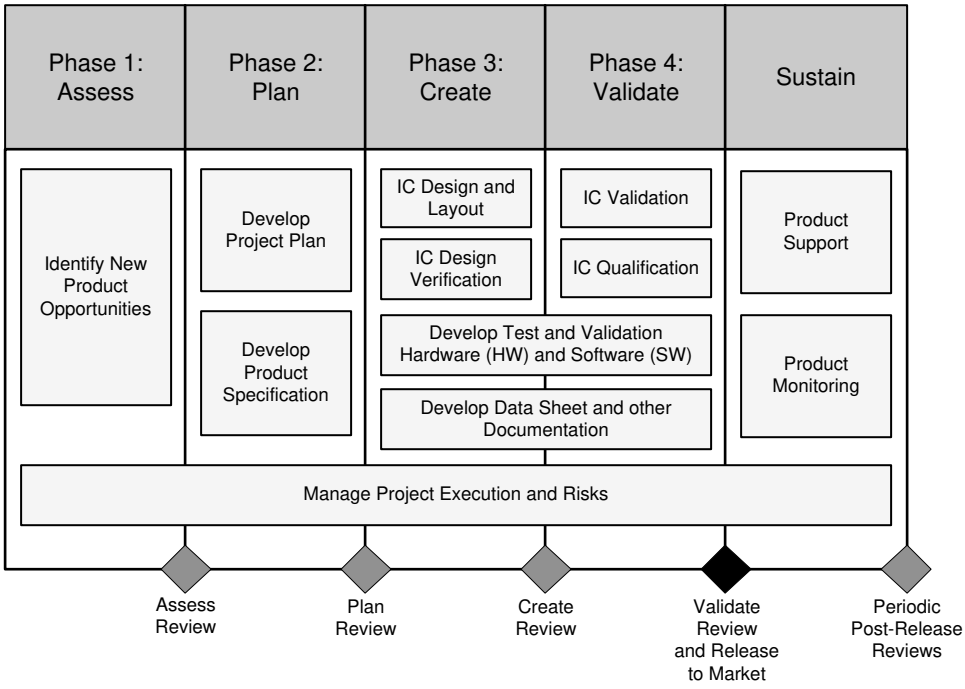

Figure 3-1. TI New-Product Development Process

3.2 TI Functional Safety Development Process

The TI functional safety development flow derives from ISO 26262 and IEC 61508 a set of requirements and methodologies to be applied to semiconductor development. This flow is combined with TI's standard new product development process to develop TI functional safety components. The details of this functional safety development flow are described in the TI internal specification - SafeTI Functional Safety Hardware.

Key elements of the TI functional safety-development flow are as follows:

- Assumptions on system level design, functional safety concept, and requirements based on TI's experience with components in functional safety applications
- Qualitative and quantitative functional safety analysis techniques including analysis of silicon failure modes and application of functional safety mechanisms
- Base FIT rate estimation based on multiple industry standards and TI manufacturing data
- Documentation of functional safety work products during the component development
- Integration of lessons learned through multiple functional safety component developments, functional safety standard working groups, and the expertise of TI customers

Table 3-1 lists these functional safety development activities which are overlaid atop the standard development flow in Figure 3-1.

Refer to Appendix B for more information about which functional safety lifecycle activities TI performs.

The customer facing work products derived from this TI functional safety process are applicable to many other functional safety standards beyond ISO 26262 and IEC 61508.

Table 3-1. Functional Safety Activities Overlaid on top of TI's Standard Development Process

Assess	Plan	Create	Validate	Sustain and End-of-Life
Determine if functional safety process execution is required	Define component target SIL/ASIL capability	Develop component level functional safety requirements	Validate functional safety design in silicon	Document any reported issues (as needed)
Nominate a functional safety manager	Generate functional safety plan	Include functional safety requirements in design specification	Characterize the functional safety design	Perform incident reporting of sustaining operations (as needed)
End of Phase Audit	Verify the functional safety plan	Verify the design specification	Qualify the functional safety design (per AEC-Q100)	Update work products (as needed)
	Initiate functional safety case	Start functional safety design	Finalize functional safety case	
	Analyze target applications to generate system level functional safety assumptions	Perform qualitative analysis of design (i.e. failure mode analysis)	Perform assessment of project	
	End of Phase Audit	Verify the qualitative analysis	Release functional safety manual	
		Verify the functional safety design	Release functional safety analysis report	
		Perform quantitative analysis of design (i.e. FMEDA)	Release functional safety report	
		Verify the quantitative analysis	End of Phase Audit	
		Iterate functional safety design as necessary		
		End of Phase Audit		

4 MSPM0L Component Overview

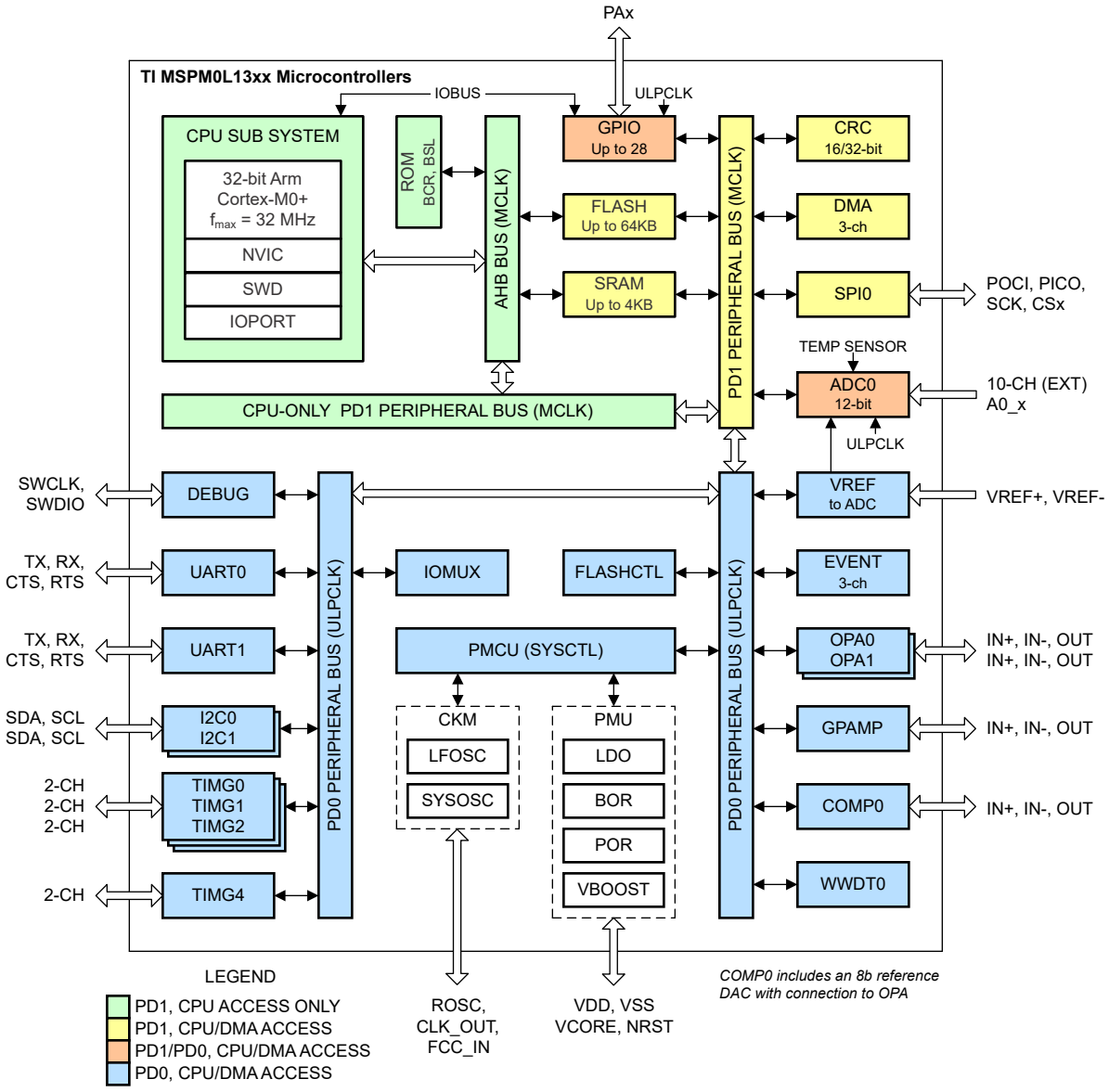

Figure 4-1. MSPM0L Block Diagram

4.1 Targeted Applications

The MSPM0L component is targeted at general-purpose functional safety applications. This is called Safety Element out of Context (SEooC) development according to ISO 26262-10. In this case, the development is done based on assumptions on the conditions of the semiconductor component usage, and then the assumptions are verified at the system level. This method is also used to meet the related requirements of IEC 61508 at the semiconductor level. This section describes some of the target applications for this component, the component safety concept, and then describes the assumptions about the systems (also know as Assumptions of Use or AoU) that were made in performing the safety analysis.

Example target applications include, but are not limited to, the following:

- Automotive - Person occupancy detection
- Automotive - Lighting
- Automotive - Seat heaters
- Automotive - Window control

MSPM0L Typical application shows a generic block diagram for applications listed above. This diagram is only an example and may not represent a complete system.

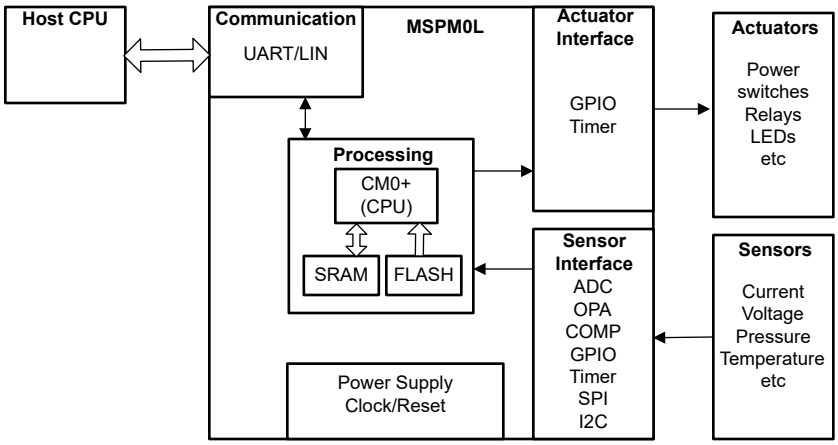

Figure 4-2. Typical Application

4.2 Hardware Component Functional Safety Concept

In case of internal errors in the component, one or more of the following actions can be taken:

- Take the actuator output pins to a safe state (For example the fault logic in timers can take the outputs to a safe state).
- Reset the device.
- Communicate the error to the host CPU and let system take the appropriate action.

4.3 Functional Safety Constraints and Assumptions

In creating a functional Safety Element out of Context (SEooC) concept and doing the functional safety analysis, TI generates a series of assumptions on system level design, functional safety concept, and requirements. These assumptions (sometimes called Assumptions of Use) are listed below. Additional assumptions about the detailed implementation of safety mechanisms are separately located in Section 6.3.

The MSPM0L Functional Safety Analysis was done under the following system assumptions:

- **[SA_1]** The system integrator shall follow all requirements in the component data sheet.

During integration activities these assumptions of use and integration guidelines described for this component shall be considered. Use caution if one of the above functional safety assumptions on this component cannot be met, as some identified gaps may be unresolvable at the system level.

5 Description of Hardware Component Parts

A semiconductor component can be divided into parts to enable a more granular functional safety analysis. This can be useful to help assign specific functional safety mechanisms to portions of the design where they provide coverage ending up with a more complete and customizable functional safety analysis. This section includes a brief description of each hardware part of this component and lists the functional safety mechanisms that can be applied to each. This section is intended to provide additional details about the assignment of functional safety mechanisms that can be found in the Safety Analysis Report. The content in this section is also summarized in Appendix A.

5.1 ADC

The 12-bit analog-to-digital converter (ADC) module in these devices support fast 12-bit conversions with single-ended inputs.

ADC features include:

- 12-bit output resolution at up to 1.68 Msps with greater than 11-bit ENOB
- HW averaging enables 14-bit conversion resolution at 105ksps
- Up to 10 external input channels
- Internal channels for temperature sensing, supply monitoring, and analog signal chain (interconnection with OPA, GPAMP, and others)
- Software selectable reference:
  - Configurable internal dedicated ADC reference voltage of 1.4 V and 2.5 V (VREF)
  - MCU supply voltage (VDD)
  - External reference supplied to the ADC through the VREF+ and VREF- pins
- Operates in RUN, SLEEP, and STOP modes and supports triggers from STANDBY mode

For more details, see the ADC chapter of the *MSPM0 L-Series 32-MHz Microcontrollers Technical Reference Manual*.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- ADC1
- ADC2
- ADC3
- ADC4
- WDT

5.2 Comparator

The comparator peripheral in the device compares the voltage levels on two inputs terminals and provides a digital output based on this comparison. The COMP supports the following key features:

- Programmable hysteresis
- Programmable reference voltage:
  - Integrated 8-bit reference DAC, the output can also can connect to OPA input terminal internally as an output buffer.
- Configurable operation modes:
  - High-speed mode (for the lowest propagation delay in timing-critical applications)
  - Low-power mode (for monitoring slow-moving signals at the lowest power consumption)
- Programmable output glitch filter delay
- Support output wake up device from all low-power modes
- Output connected to advanced timer fault handling mechanism
- The IPSEL and IMSEL bits in comparator registers can be used to select the comparator channel inputs from device pins or from internal analog modules.

For more details, see the COMP chapter of the *MSPM0 L-Series 32-MHz Microcontrollers Technical Reference Manual*.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- COMP1
- COMP3

5.3 OPA

The zero-drift op amps (OPAs) in these devices, OPA0 and OPA1, are chopper stabilized operational amplifiers with rail-to-rail input/output and a programmable gain stage feedback loop.

The OPA peripherals support the following key features:

- Rail-to-rail input and output
- Dual power modes for added flexibility
- Configurable amplifier modes include General Purpose Mode, Buffer Mode, Inverting/Non-inverting PGA Mode, Difference Amplifier Mode and Cascade Amplifier Mode
- Software-selectable zero-drift chopper stabilization for improved accuracy and drift performance
- MSPM0L134x devices feature dedicated OPA inverting input pins for 10 pA I_b performance
- Factory trimming to remove offset error
- Burnout current source (BCS) integrated to monitor sensor health
- Programmable gain amplifier (PGA) up to 32x

The OPA features configurable input muxes P-MUX, N-MUX, and M-MUX to support various analog signal chain amplifier configurations that include general purpose, inverting, noninverting, unity gain, cascade, noninverting cascade, difference, and more.

For more details, see the OPA chapter of the *MSPM0 L-Series 32-MHz Microcontrollers Technical Reference Manual*.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- OA1
- OA2

5.4 CPU

The CPU subsystem (MCPUSS) implements an Arm Cortex-M0+ CPU, an instruction prefetch and cache, a system timer, and interrupt management features. The Arm Cortex-M0+ is a cost-optimized 32-bit CPU that delivers high performance and low power to embedded applications. Key features of the CPU Sub System include:

- Arm Cortex-M0+ CPU supports clock frequencies from 32 kHz to 32 MHz
  - ARMv6-M Thumb instruction set (little endian) with single-cycle 32×32 multiply instruction
  - Single-cycle access to GPIO registers through Arm single-cycle IO port
- Prefetch logic to improve sequential code execution, and I-cache with 2 64-bit cache lines
- System timer (SysTick) with 24-bit down counter and automatic reload
- Nested vectored interrupt controller (NVIC) with 4 programmable priority levels and tail chaining
- Interrupt groups for expanding the total interrupt sources, with jump index for low interrupt latency

For more details, see the CPU chapter of the *MSPM0 L-Series 32-MHz Microcontrollers Technical Reference Manual*.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- CPU1

5.5 RAM

MSPM0 L-series MCUs (MSPM0Lxx) include a low-power high-performance SRAM memory with zero wait state access across the supported CPU frequency range of the device. SRAM memory can be used for storing volatile

information such as the call stack, heap, global data, and code. The SRAM memory content is fully retained in run, sleep, stop, and standby operating modes and is lost in shutdown mode. A write protection mechanism is provided to allow the application to prevent unintended modifications to a portion of the SRAM memory. SRAM write protection is useful when placing executable code into SRAM to provide a level of protection against unintentional overwrites of code by either the CPU or DMA. Placing code in SRAM can improve performance of critical loops by enabling zero wait state operation and lower power consumption.

For more details, see the "Platform memory map" chapter of the *MSPM0 L-Series 32-MHz Microcontrollers Technical Reference Manual*.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- SYSMEM1
- SYSMEM2
- SYSMEM5

5.6 FLASH

A single bank of nonvolatile flash memory is provided for storing executable program code and application data.

Key features of the flash include:

- In-circuit program and erase operations supported across the entire recommended supply range
- Small 1KB sector sizes (minimum erase resolution of 1KB)
- Up to 100000 program and erase cycles on the lower 32KB of the flash memory, with up to 10000 program and erase cycles on the remaining flash memory (devices with 32KB or less support 100000 cycles on the entire flash memory)

For a complete description of the flash memory, see the NVM chapter of the *MSPM0 L-Series 32-MHz Microcontrollers Technical Reference Manual*.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- FLASH2

5.7 GPIO

The general purpose input/output (GPIO) peripheral lets the application write data out and read data in through the device pins. Through the use of the Port A GPIO peripheral, these devices support up to 28 GPIO pins.

The key features of the GPIO module include:

- 0 wait state MMR access from CPU
- Set, clear, or toggle multiple bits without the need of a read-modify-write construct in software
- "FastWake" feature enables low-power wakeup from STOP and STANDBY modes for any GPIO port
- User controlled input filtering

For more details, see the GPIO chapter of the *MSPM0 L-Series 32-MHz Microcontrollers Technical Reference Manual*.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- GPIO1
- GPIO2

5.8 DMA

The direct memory access (DMA) controller allows movement of data from one memory address to another without CPU intervention. For example, the DMA can be used to move data from ADC conversion memory to SRAM. The DMA reduces system power consumption by allowing the CPU to remain in low power mode, without having to awaken to move data to or from a peripheral.

The DMA in these devices support the following key features:

- 3 independent DMA transfer channels
  - 1 full-feature channel (DMA0), supporting repeated transfer modes
  - 2 basic channels (DMA1, DMA2), supporting single transfer modes
- Configurable DMA channel priorities
- Byte (8-bit), short word (16-bit), word (32-bit) and long word (64-bit) or mixed byte and word transfer capability
- Transfer counter block size supports up to 64k transfers of any data type
- Configurable DMA transfer trigger selection
- Active channel interruption to service other channels
- Early interrupt generation for ping-pong buffer architecture
- Cascading channels upon completion of activity on another channel
- Stride mode to support data re-organization

For more details, see the DMA chapter of the *MSPM0 L-Series 32-MHz Microcontrollers Technical Reference Manual*.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- DMA1
- DMA2

5.9 SPI

The serial peripheral interface (SPI) peripherals in these devices support the following key features:

- Support ULPCLK/2 bit rate and up to 16 Mbits/s in both controller and peripheral mode
- Configurable as a controller or a peripheral
- Configurable chip select for both controller and peripheral
- Programmable clock prescaler and bit rate
- Programmable data frame size from 4-bits to 16-bits (Controller Mode)
- Programmable data frame size from 7-bits to 16-bits (Peripheral Mode)
- Separated transmit and receive FIFOs support DMA data transfer
- Supports TI mode, Motorola mode and National Microwire format

For more details, see the SPI chapter of the *MSPM0 L-Series 32-MHz Microcontrollers Technical Reference Manual*.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- SPI1
- SPI2
- SPI3

5.10 I2C

The inter-integrated circuit interface (I^2C) peripherals in these devices provide bidirectional data transfer with other I2C devices on the bus and support the following key features:

- 7-bit and 10-bit addressing mode with multiple 7-bit target addresses
- Multiple-controller transmitter or receiver mode
- Target receiver or transmitter mode with configurable clock stretching
- Support Standard-mode (Sm), with a bit rate up to 100 kbit/s
- Support Fast-mode (Fm), with a bit rate up to 400 kbit/s
- Support Fast-mode Plus (Fm+), with a bit rate up to 1 Mbit/s
- Separated transmit and receive FIFOs support DMA data transfer
- Support SMBus 3.0 with PEC, ARP, timeout detection and host support
- Wakeup from low power mode on address match
- Support analog and digital glitch filter for input signal glitch suppression

For more details, see the I2C chapter of the *MSPM0 L-Series 32-MHz Microcontrollers Technical Reference Manual*.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- I2C1
- I2C2

5.11 UART

The UART peripherals provide the following key features:

- Standard asynchronous communication bits for start, stop, and parity
- Fully programmable serial interface
  - 5, 6, 7 or 8 data bits
  - Even, odd, stick, or no-parity bit generation and detection
  - 1 or 2 stop bit generation
  - Line-break detection
  - Glitch filter on the input signals
  - Programmable baud rate generation with oversampling by 16, 8 or 3
  - Local Interconnect Network (LIN) mode support
- Separated transmit and receive FIFOs support DAM data transfer
- Support transmit and receive loopback mode operation
- See UART Features for detail information on supported protocols

Table 5-1. UART Features

UART Features	UART0 (Extend)	UART1 (Main)
Active in Stop and Standby Mode	Yes	Yes
Separate transmit and receive FIFOs	Yes	Yes
Support hardware flow control	Yes	Yes
Support 9-bit configuration	Yes	Yes
Support LIN mode	Yes	-
Support DALI	Yes	-
Support IrDA	Yes	-
Support ISO7816 Smart Card	Yes	-
Support Manchester coding	Yes	-
FIFO Depth	4 entries	4 entries

For more details, see the UART chapter of the MSPM0L-Series 32-MHz Microcontrollers Technical Reference Manual.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- UART1
- UART2

5.12 Timers (TIMx)

The timer peripherals in these devices support the following key features. For specific configuration, see Different TIMG Configurations.

Specific features for the **general-purpose timer (TIMGx)** include:

- 16-bit down, up/down, or up counter with repeat-reload mode
- Selectable and configurable clock source
- 8-bit programmable prescaler to divide the counter clock frequency
- Two independent channels for
  - Output compare

  - Input capture
  - PWM output
  - One-shot mode
- Support quadrature encoder interface (QEI) for positioning and movement sensing
- Support synchronization and cross trigger among different TIMx instances in the same power domain
- Support interrupt/DMA trigger generation and cross peripherals (such as ADC) trigger capability
- Cross-trigger event logic for Hall sensor inputs

Table 5-2. Different TIMG Configurations

TIM Name	Power Domain	Resolution	Prescaler	Capture/ Compare Channels	External PWM Channels	Phase Load	Shadow Load	Shadow CC
TIMG0	PD0	16-bit	8-bit	2	2	-	-	-
TIMG1	PD0	16-bit	8-bit	2	2	-	-	-
TIMG2	PD0	16-bit	8-bit	2	2	-	-	-
TIMG4	PD0	16-bit	8-bit	2	2	-	Yes	Yes

For more details, see the timer chapters of the *MSPM0 L-Series 32-MHz Microcontrollers Technical Reference Manual*.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- TIM1
- TIM2

5.13 PMU

The power management unit (PMU) generates the internally regulated core supplies for the device and provides supervision of the external supply (VDD). The PMU also contains the bandgap voltage reference used by the PMU itself as well as analog peripherals. Key features of the PMU include:

- Power-on reset (POR) supply monitor
- Brown-out reset (BOR) supply monitor with early warning capability using three programmable thresholds
- Core regulator with support for RUN, SLEEP, STOP, and STANDBY operating modes to dynamically balance performance with power consumption
- Parity-protected trim to immediately generate a power-on reset (POR) in the event that a power management trim is corrupted

For more details, see the PMU chapter of the *MSPM0 L-Series 32-MHz Microcontrollers Technical Reference Manual*.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- SYSCTL8
- SYSCTL10
- SYSCTL14
- SYSCTL15

5.14 CKM

The clock module provides the following oscillators:

- **LFOSC**: Internal low-frequency oscillator (32 kHz)
- **SYSOSC**: Internal high-frequency oscillator (4 MHz or 32 MHz with factory trim, 16 MHz or 24 MHz with user trim)

The following clocks are distributed by the clock module for use by the processor, bus, and peripherals:

- **MCLK**: Main system clock for PD1 peripherals, derived from SYSOSC or LFCLK, active in RUN and SLEEP modes
- **CPUCLK**: Clock for the processor (derived from MCLK), active in RUN mode
- **ULPCLK**: Ultra-low power clock for PD0 peripherals, active in RUN, SLEEP, STOP, and STANDBY modes
- **MFCLK**: 4-MHz fixed mid-frequency clock for peripherals, available in RUN, SLEEP, and STOP modes
- **LFCLK**: 32-kHz fixed low-frequency clock for peripherals or MCLK, active in RUN, SLEEP, STOP, and STANDBY modes
- **ADCCLK**: ADC clock, available in RUN, SLEEP and STOP modes
- **CLK_OUT**: Used to output a clock externally, available in RUN, SLEEP, STOP, and STANDBY modes

For more details, see the CKM chapter of the *MSPM0 L-Series 32-MHz Microcontrollers Technical Reference Manual*.

The following tests can be applied as functional safety mechanisms for this module (to provide diagnostic coverage on a specific function):

- SYSCTL1
- SYSCTL5
- SYSCTL9
- SYSCTL11
- SYSCTL12

6 MSPM0L Management of Random Faults

For a functional safety critical development it is necessary to manage both systematic and random faults. The MSPM0L component architecture includes many functional safety mechanisms, which can detect and respond to random faults when used correctly. This section of the document describes the architectural functional safety concept for each sub-block of the MSPM0L component. The system integrator shall review the recommended functional safety mechanisms in the functional safety analysis report (FMEDA) in addition to this safety manual to determine the appropriate functional safety mechanisms to include in their system. The component data sheet or technical reference manual (if available) are useful tools for finding more specific information about the implementation of these features.

6.1 Fault Reporting

Internal faults are reported to the host controller through the host bus.

6.2 Functional Safety Mechanism Categories

This section includes a description of the different types of functional safety mechanisms that are applied to the design blocks of the MSPM0L component.

The functional safety mechanism categories are defined as follows:

Component Hardware Functional Safety Mechanisms	A safety mechanism that is implemented by TI in silicon which can communicate error status upon the detection of failures. The safety mechanism may require software to enable its functionality, to take action when a failure is detected, or both.
Component Hardware and Software Functional Safety Mechanisms	A test recommended by TI which requires both, safety mechanism hardware which has been implemented in silicon by TI, and which requires software. The failure modes of the hardware used in this safety mechanisms are analyzed or described as part of the functional safety analysis or FMEDA. The system implementer is responsible for analyzing the software aspects for this safety mechanism.
Component Software Functional Safety Mechanisms	A software test recommended by TI. The failure modes of the software used in this safety mechanism are not analyzed or described in the functional safety analysis or FMEDA. For some components, TI may provide example code or supporting code for the software functional safety mechanisms. This code is intended to aid in the development, but the customer shall do integration testing and verification as needed for their system functional safety concept.
System Functional Safety Mechanisms	A safety mechanism implemented externally of this component. For example an external monitoring IC would be considered to be a system functional safety mechanism.
Test for Safety Mechanisms	This test provides coverage for faults on a safety mechanism only. It does not provide coverage for the primary function.
Alternative Safety Mechanisms	An alternative safety mechanism is not capable of detecting a fault of safety mechanism hardware, but instead is capable of recognizing the primary function fault (that another safety mechanism may have failed to detect). Alternate safety mechanisms are typically used when there is no direct test for a safety mechanism.

6.3 Description of Functional Safety Mechanisms

This section provides a brief summary of the functional safety mechanisms available on the various components in this device.

6.3.1 ADC1,DMA1,COMP1,GPIO2,TIM2,I2C2,IOMUX1,OA1,SPI2,UART2,SYSCTL5,REF1: Periodic read of static configuration registers

Periodic software read of static configuration registers is one of the methods listed in ISO26262. This mechanism involves the software verifying the values of static configuration registers against a known reference. One of the methods is to compute a CRC signature for the static configuration register values and periodically computing the CRC and checking against expected CRC signature.

6.3.2 ADC2: Software test of function

In this method, internal DAC8(COMPDAC) output is used to setup a known voltage on the ADC input channel. As there is no direct connection between DAC8 in the comparator and the ADC, DAC8 has to be connected to ADC through the OPA. OPA can be configured as a buffer. Other parameters of the ADC like the sampling time, reference sources, sequencer modes are setup similar to the actual application. Software trigger can be utilized to trigger the ADC. The output of the ADC can be compared against the expected range.

6.3.3 ADC3: ADC trigger overflow check

If a trigger to ADC is fired, while a sample/conversion operation is in progress, TOVIFG flag will be set. This flag can be configured to generate an interrupt and appropriate action can be taken.

6.3.4 ADC4: Window comparator

There is one window comparator unit available in the ADC which can be used to check if the input signal is within predefined threshold values set by software. The ADC result that goes into MEMRES or FIFO is what gets checked against the threshold values of the window comparator.

Based on the comparison it can generate 3 interrupt conditions:

1. LOWIFG – Conversion result is below the Low threshold (WCLOW)
2. HIGHIFG – Conversion result is above the High threshold (WCHIGH)
3. INIFG – Conversion result is in between or equal to the Low and High thresholds

The window comparator low and high threshold values are global for all channels and the window comparison feature can be enabled for each channel as needed using the WINCOMP bit in the MEMCTL register.

When the ADC result data format (CTL2.DF) or resolution (CTL2.RES) configuration is changed, the window comparator threshold values are not reset by hardware and are retained as is. The software application is expected to reconfigure the threshold values as appropriate after changing the data format and/or resolution configuration.

6.3.5 OA2: Test of OA using internal DAC8 as a driver

In this test method, DAC8 (in the comparator) output is chosen as input to OPA. The OPA is configured as per the application configuration. The output of OPA can be measured using ADC.

6.3.6 COMP3: Testing COMP using an external pin

In this method, external pin is connected to the comparator positive terminal input and negative terminal is fed by internal DAC8(COMPDAC). Required test voltage is driven on the external pin. Software can check whether the output from COMP is as expected.

6.3.7 CPU1: CPU test using software test library

The CortexM0+ CPU and MPU will be tested using the ARM Software Test Library (STL).

6.3.8 DMA2: Software test of DMA function

In this test mechanism, one of the DMA channels is dedicated to diagnostic test. This channel can be configured to do transfers of known data content from a fixed source (SRAM/FLASH) to a fixed destination (SRAM/CRC engine). Periodically the diagnostic channel can be triggered in software and the proper transfer of data can be checked in software.

6.3.9 SYSMEM1: Write to SRAM from CPU, read from DMA

In this test a fixed data can be written to SRAM from CPU and the DMA can be used to transfer this data to the CRC for CRC computation.

6.3.10 SYSMEM2: Write to SRAM from DMA, read from CPU

In this test a fixed data can be written to SRAM from DMA and CPU can be used to transfer this data to the CRC for CRC computation.

6.3.11 SYSMEM5: SRAM March test

A March test consists of a finite sequence of March elements; while a March element is a finite sequence of operations applied to every cell in the memory array before proceeding to the next cell. For example, an operation can consist of writing a 0 into a cell, writing a 1 into a cell, reading an expected 0 from a cell, and reading an expected 1 from a cell. A failure is detected if the expected “1” is not read. The coverage level for linked cells depends on the write/read order.

For more information regarding march tests, refer to A. J. Van De Goor, "Using march tests to test SRAMs," in IEEE Design & Test of Computers, vol. 10, no. 1, pp. 8-14, March 1993, doi: 10.1109/54.199799.

6.3.12 FXBAR1: CPU readback of known data from Flash

In this test, known data is read back from CPU spread across whole flash region. The number of data points can be a small subset of the all the flash locations. Potentially a CRC check can be done to check the correctness of the data read.

6.3.13 FXBAR2: DMA readback of known data from Flash

In this test, known data is read back from DMA spread across whole flash region. The number of data points can be a small subset of the all the flash locations. Potentially a CRC check can be done to check the correctness of the data read.

6.3.14 FLASH2: CRC check of flash content

A boot or during run time (background crc check), a crc check of the flash content can be done to ensure that the flash content is not corrupted.

6.3.15 GPIO1: GPIO test using pin IO loopback

In this test method, the GPIO pin is setup with a known value and the status of the pin can be read back using the DIN register.

6.3.16 WDT

The windowed watchdog timer (WWDT) can be used to supervise the operation of the device, specifically code execution. The WWDT can be used to generate a reset or an interrupt if the application software does not successfully reset the watchdog within a specified window of time. Key features of the WWDT include:

- 25-bit counter
- Programmable clock divider
- Eight software selectable watchdog timer periods
- Eight software selectable window sizes
- Support for stopping the WWDT automatically when entering a sleep mode
- Interval timer mode for applications which do not require watchdog functionality

For more details, see the WWDT chapter of the *MSPM0 L-Series 32-MHz Microcontrollers Technical Reference Manual*.

6.3.17 TIM1: Software test of function

In this test mechanism, the expected PWM waveform is generated on one timer (Timer 1) and the signal is looped back on the board to another timer's (Timer 2) capture input. The various timing parameters of the waveform are measured using Timer 2.

6.3.18 I2C1: Software test of I2C function using internal loopback mechanism

The I^2C modules can be placed into an internal loopback mode for diagnostic or debug work by setting the LPBK bit in the I^2C controller configuration I2Cx.MCR register. In loopback mode, the SDA and SCL signals from the controller part of the I^2C are tied to the SDA and SCL signals of the target part of the I^2C module to allow internal testing of the device without having to connect the I/Os.

This loopback mechanism can be used to transmit known data from transmit to receive. The I2C configuration can be similar to the application configuration with respect to the bit rate, FIDO usage etc. The completion of the test can be timed to be within expected limits to detect any faults in the bit rate timing.

6.3.19 SPI1 : Software test of SPI function

The SPI can be placed into an internal loopback in Controller mode for by setting the LBM bit in the SPI.CTL1 register. In loopback mode, data transmitted on the TX output is received on the RX input. Application can use FIFO to check out FIFO behaviour. For checking the clock setting, the test execution time can be timed using timers.

6.3.20 SPI3: SPI periodic safety message exchange

An application level check can be added to periodically send and receive test message when SPI is in peripheral mode. In software, timeout mechanism needs to be implemented to cover for this.

6.3.21 UART1: Software test of UART function

The UART can be placed into an internal loopback mode for diagnostic or debug work by setting the LBE bit in the UART.CTL0 register. In loopback mode, data transmitted on the TXD output is received on the RXD input. Data received on the RXD IO pin will be ignored when loopback is enabled. Use an SoC timer/ watch dog to indicate if the communication is completed within the expected amount of time.

6.3.22 SYSCTL1: MCLK monitor

A digital clock monitor is provided to ensure that MCLK is alive. An MCLK fault is asserted by the MCLK monitor if there is no MCLK activity within a period of 1-12 LFCLK cycles. An MCLK fault is always considered fatal to the system and will generate a BOOTRST.

The MCLK monitor can be enabled once LFCLK is configured and running. To enable the MCLK monitor, set the MCLKDEADCHK bit in the MCLKCFG register in SYSCTL. When enabled, the MCLK monitor runs in all operating modes except for STANDBY1 and SHUTDOWN.

6.3.23 SYSCTL8: Brownout Reset (BOR) Supervisor

The brownout reset (BOR) supervisor monitors the external supply (VDD) and asserts or de-asserts a BOR violation to SYSCTL.

6.3.24 SYSCTL9: FCC counter logic to calculate clock frequencies

The frequency clock counter (FCC) enables flexible in-system testing and calibration of a variety of oscillators and clocks on the device. The FCC counts the number of clock periods seen on the selected source clock within a known trigger period (derived from a secondary reference source) to provide an estimation of the frequency of the source clock.

6.3.25 SYSCTL10: External voltage monitor

Use an external voltage monitor on VCORE pin to monitor the LDO output.

6.3.26 SYSCTL11: Boot process monitor

In the event of a boot fail during execution of the boot configuration routine (BCR), SYSCTL will assert a BOOTRST to re-attempt a successful boot. A boot fail can be caused by any of the following:

1. Boot configuration data integrity error (This can be used for BOOTROM/ FLASH)
2. Device trim integrity error (This can be used for FLASH)
3. BCR timeout (BCR takes significantly longer than expected to complete for any other reason) (This can be used for BOOTROM)

6.3.27 SYSCTL12: Shutdown memory bits parity protection

The SoC has a couple of registers (SHUTDNSTOREx) which retain their content during SHUTDOWN mode. Since there is no active state monitoring by the CPU possible in SHUTDOWN mode, a parity protection has been added to these registers. On the detection of a parity error, the PMU will issue a POR reset request. If the device was in SHUTDOWN mode, the device will not reset right away. The device will wakeup with a next wakeup event and the PMU will issue the POR request and clear the SHUTDNSTOREx registers. The SW will see this wakeup as POR reset, rather then a wakeup from SHUTDOWN mode, which indicates the content in the SHUTDNSTOREx registers has been reset.

6.3.28 SYSCTL14: Brownout Voltage Monitor

The brownout circuit can be configured to monitor the external supply (VDD) above the BOR reset level. If tripped, the monitor will generate a none-maskable-interrupt (NMI) event and reconfigure the BOR circuit to be a BOR supervisor reset. This allows SW to either warn the user of a depleting battery or initiate a graceful shutdown of the system application, before the BOR circuit will issue a BOR supervisor reset.

6.3.29 SYSCTL15: External voltage supervisor on VDD

An external voltage supervisor can be added to the system, which can monitor the VDD line and pull the reset if the VDD is out of expected range.

6.3.30 REF2: Test of VREF using ADC

In this test, internal reference voltage given out from VREF is used as reference voltage in ADC for ADC operations and the converted result is compared against the expected digital value.

A Summary of Recommended Functional Safety Mechanism Usage

Table A-2 summarizes the functional safety mechanisms present in hardware or recommend for implementation in software or at the system level as described in Section 5. Table A-1 describes each column in Table A-2 and gives examples of what content could appear in each cell.

Table A-1. Legend of Functional Safety Mechanisms

Functional Safety Mechanism	Description
TI Safety Mechanism Unique Identifier	A unique identifier assigned to this safety mechanism for easier tracking.
Safety Mechanism Name	The full name of this safety mechanism.
Safety Mechanism Category	**Safety Mechanism** - This test provides coverage for faults on the primary function. It may also provide coverage on another safety mechanism. **Test for Safety Mechanism** - This test provides coverage for faults of a safety mechanism only. It does not provide coverage on the primary function. **Fault Avoidance** - This is typically a feature used to improve the effectiveness of a related safety mechanism.
Safety Mechanism Type	Can be either hardware, software, a combination of both hardware and software, or system. See Section 6.2 for more details.
Safety Mechanism Operation Interval	The timing behavior of the safety mechanism with respect to the test interval defined for a functional safety requirement / functional safety goal. Can be either continuous, or on-demand. **Continuous** - the safety mechanism constantly monitors the hardware-under-test for a failure condition. **Periodic or On-Demand** - the safety mechanism is executed periodically, when demanded by the application. This includes Built-In Self-Tests that are executed one time per drive cycle or once every few hours.
Test Execution Time	Time period required for the safety mechanism to complete, not including error reporting time. Note: Certain parameters are not set until there is a concrete implementation in a specific component. When component specific information is required, the component data sheet should be referenced. Note: For software-driven tests, the majority contribution of the Test Execution Time is often software implementation-dependent.
Action on Detected Fault	The response that this safety mechanism takes when an error is detected. Note: For software-driven tests, the Action on Detected Fault may depend on software implementation.
Time to Report	Typical time required for safety mechanism to indicate a detected fault to the system. Note: For software-driven tests, the majority contribution of the Time to Report is often software implementation-dependent.

Table A-2. Summary of Functional Safety Mechanisms

TI Safety Mechanism Unique Identifier	Safety Mechanism Name	Safety Mechanism Category	Safety Mechanism Type	Safety Mechanism Operation Interval	Test Execution Time	Action on Detected Fault	Time to Report
ADC1	Software test for periodic read of static configured MMRs	Safety Mechanism	Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
ADC2	ADC sample and conversion test	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
WDT	Watchdog Timeout Event	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
ADC3	ADC Trigger overflow	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
COMP1	Software Read Back of Written Configuration	Safety Mechanism	Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent

Table A-2. Summary of Functional Safety Mechanisms (continued)

TI Safety Mechanism Unique Identifier	Safety Mechanism Name	Safety Mechanism Category	Safety Mechanism Type	Safety Mechanism Operation Interval	Test Execution Time	Action on Detected Fault	Time to Report
COMP3	External pin input to COMP	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
CPU1	ARM STL	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
DMA1	Periodic Software Read Back of Static Configuration Registers	Safety Mechanism	Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
DMA2	Software test	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
FXBAR1	Use hardware redundancy by access same flash location by CPU and DMA	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
FXBAR2	Periodic Software Read Back of FLASH data	Safety Mechanism	Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
GPIO1	Software test of function using I/O loopback	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
GPIO2	Periodic Software Readback of Static Configuration Registers	Safety Mechanism	Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
TIM1	Test for basic PWM generation	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
TIM2	Periodic Software Read Back of Static Configuration Registers	Safety Mechanism	Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
TIM3	Test for fault generation	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
I2C1	Software test of function using I/O loopback	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
I2C2	Periodic Software Read Back of Static Configuration Registers	Safety Mechanism	Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
IOMUX1	Periodic Software Readback of Static Configuration Registers	Safety Mechanism	Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
OA1	Software Read Back of Written Configuration	Safety Mechanism	Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
OA2	DAC8(COMPDAC) to OA and then to ADC Loopback	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
SPI1	Software test of function using I/O loopback	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
SPI2	Periodic Software Read Back of Static Configuration Registers	Safety Mechanism	Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent

Table A-2. Summary of Functional Safety Mechanisms (continued)

TI Safety Mechanism Unique Identifier	Safety Mechanism Name	Safety Mechanism Category	Safety Mechanism Type	Safety Mechanism Operation Interval	Test Execution Time	Action on Detected Fault	Time to Report
SPI3	SPI PERIODIC Safety Message check	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
SYSCTL1	MCLK monitor	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
SYSCTL2	HFCLK Startup monitor	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
SYSCTL3	LFCLK Monitor	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
SYSCTL4	RTC Monitor	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
SYSCTL5	Periodic Software Read Back of Static Configuration Registers	Safety Mechanism	Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
SYSCTL6	SYSPLL Startup monitor	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
SYSCTL8	Brownout Reset (BOR) Supervisor	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
SYSCTL9	FCC counter logic to calculate clock frequencies	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
SYSCTL10	External voltage monitor	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
SYSCTL11	Boot process monitor	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
SYSCTL12	Parity protection	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
SYSCTL13	SYSCTL3V State machine	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
SYSCTL14	Brownout Voltage Monitor	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
SYSCTL15	External Voltage Supervisor	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
SYSMEM1	Software read of memory	Software	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
SYSMEM2	Software read of memory	Software	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
SYSMEM5	RAM March Test	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
UART1	Software test of function using I/O loopback	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
UART2	Periodic Software Read Back of Static Configuration Registers	Safety Mechanism	Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
REF1	Periodic Software Read Back of static configuration registers.	Safety Mechanism	Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent
REF2	VREF to ADC Reference input	Safety Mechanism	Hardware/ Software	Periodic/On-Demand	Application dependent	Reset the device	Application dependent

B Distributed Developments

A Development Interface Agreement (DIA) is intended to capture the agreement between two parties towards the management of each party's responsibilities related to the development of a functional safety system. TI functional safety components are typically designed for many different systems and are considered to be Safety Elements out of Context (SEooC) hardware components. The system integrator is then responsible for taking the information provided in the hardware component safety manual, safety analysis report and safety report to perform system integration activities. Because there is no distribution of development activities, TI does not accept DIAs with system integrators.

TI functional safety components are products that TI represents, promotes or markets as helping customers mitigate functional safety related risks in an end application and/or as compliant with an industry functional safety standard or FS-QM. For more information about TI functional safety components, go to TI.com/functionalsafety.

B.1 How the Functional Safety Lifecycle Applies to TI Functional Safety Products

TI has tailored the functional safety lifecycles of ISO 26262 and IEC 61508 to best match the needs of a functional Safety Element out of Context (SEooC) development. The functional safety standards are written in the context of the functional safety systems, which means that some requirements only apply at the system level. Since TI functional safety components are hardware or software components, TI has tailored the functional safety activities to create new product development processes for hardware and for software that makes sure state-of-the-art techniques and measures are applied as appropriate. These new product development processes have been certified by third-party functional safety experts. To find these certifications, go to TI.com/functionalsafety.

B.2 Activities Performed by Texas Instruments

The TI functional safety products are hardware components developed as functional Safety Elements out of Context. As such, TI's functional safety activities focus on those related to management of functional safety around hardware component development. System level architecture, design, and functional safety analysis are not within the scope of TI activities and are the responsibility of the customer. Some techniques for integrating the SEooC safety analysis of this hardware component into the system level can be found in ISO 26262-11.

Table B-1. Activities Performed by Texas Instruments versus Performed by the customer

Functional Safety Lifecycle Activity(1)	TI Execution	Customer Execution
Management of functional safety	Yes	Yes
Definition of end equipment and item	No	Yes
Hazard analysis and risk assessment (of end equipment/item)	No	Yes
Creation of end equipment functional safety concept	No. Assumptions made for internal development.	Yes
Allocation of end equipment requirements to sub-systems, hardware components, and software components	No. Assumptions made for internal development.	Yes
Definition of hardware component safety requirements	Yes	No
Hardware component architecture and design execution	Yes	No
Hardware component functional safety analysis	Yes	No
Hardware component verification and validation (V&V)	V&V executed to support internal development.	Yes
Integration of hardware component into end equipment	No	Yes
Verification of IC performance in end equipment	No	Yes
Selection of safety mechanisms to be applied to IC	No	Yes
End equipment level verification and validation	No	Yes
End equipment level functional safety analysis	No	Yes
End equipment level functional safety assessment	No	Yes
End equipment release to production	No	Yes
Management of functional safety issues in production	Support provided as needed	Yes

(1) For component technical questions, ask our TI E2E™ support experts.

B.3 Information Provided

Texas instruments has summarized what it considers the most critical functional safety work products that are available to the customer either publicly or under a nondisclosure agreement (NDA). NDAs are required to protect proprietary and sensitive information disclosed in certain functional safety documents.

Table B-2. Product Functional Safety Documentation

Deliverable Name	Contents
Functional Safety Product Preview	Overview of functional safety considerations in product development and product architecture. Delivered ahead of public product announcement.
Functional Safety Manual	User guide for the functional safety features of the product, including system level assumptions of use.
Functional Safety Analysis Report	Results of all available functional safety analysis documented in a format that allows computation of custom metrics.
Functional Safety Report(1)	Summary of arguments and evidence of compliance to functional safety standards. References a specific component, component family, or TI process that was analyzed.
Assessment Certificate(1)	Evidence of compliance to functional safety standards. References a specific component, component family, or TI process that was analyzed. Provided by a 3rd party functional safety assessor.

(1) When an Assessment Certificate is available for a TI functional safety product, the Functional Safety Report may not be provided. When a Functional Safety Report is provided, an Assessment Certificate may not be available. These two documents fulfill the same functional safety requirements and will be used interchangeably depending on the TI functional safety product.

C Revision History

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

DATE	REVISION	NOTES
December 2023	*	Initial release

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