AN4475
Application note

Safety Application Guide for SPC563Mxx family

Introduction

This document contains guidelines in order to configure and use the SPC563Mxx device for safety relevant applications. These guidelines are preceded by one of the following bold text statements:

- Suggested
- Implementation hint
- Rationale

These guidelines are useful approaches for the specific topics under discussion even if they are not mandatory. The user needs to use discretion in deciding whether these measures are appropriate for their applications.

This document is valid only under the assumption that the MCU is used in automotive applications for use cases requiring a fail-safe MCU and if the environmental conditions specified in the SPC563Mxx device data sheet are maintained.

Together with the standard documentation as the reference manual and the datasheet, also SPC563Mxx device errata sheet must be taken into account during system design and implementation (see *Chapter Appendix B: Document references*).

Contents

List of tables

1 General information

1.1 Mission profile

The assumed mission profile is:

- Lifetime: 20 years
- Total operating hours: 12000 hours
- Trip time[(a)]: 10 hours
- Fault Tolerant Time Interval[(b)]: 10 ms

1.2 Safe state

The Safe states of the SPC563MXX are as follows:

- Completely unpowered
- Reset
- Operating correctly
- Explicitly indicating an internal error

If the SPC563Mxx signals an internal failure via its error out signals, the surrounding subsystem is longer use the SPC563Mxx outputs for safety functions since these signals are no longer considered reliable. If an error is indicated, the system must be able to remain in a Safe state without any additional MCU actions. Depending on its configuration, the system may disable, or reset, the SPC563Mxx as a reaction to the error signal.

Suggested: the system must set the system itself to a safe state when an error is indicated.

1.3 Failure indication time

The SPC563Mxx failure indication time has taken into consideration when determining application safety strategies, because it must be less than the FTTI.

1.4 Error handling

Error handling can be split into two categories:

- Handling of errors during run-time
- Handling of errors during boot-time

Suggested: run-time failures are handled in a time shorter than the FTTI.

Suggested: boot-time failures are handled before the safety function starts.

a. Trip time is defined as the maximum MCU operation time without Power-On Reset.

b. Quoting the ISO262562, the Fault Tolerant Time Interval (FTTI) is the time-span in which a fault or faults can be present in a system before a hazardous event occurs.

2 Functional safety requirements for application software

This section gives an overview of suggested measures when using the individual modules of the SPC563Mxx.

It is possible to ignore aspects of the text if equivalent measures are considered to manage the same failures. Modules not explicitly covered by this document are assumed not safety relevant and do not require any software measures.

2.1 Application software requirements

Application software is developed according to safety requirements.

The following sections contain **suggested** assumptions and requirements for using the SPC563Mxx devices in a safety application.

2.2 Core

Suggested: [covers: HW_CORE_EXCEPTION] all exception are enabled, if not enabled by default, and managed. These specific software countermeasures can run once after the Power-On Reset (POR) before the safety function starts. [end]

Suggested: [covers: SW_CODE_REDUND] this safety mechanism consists of two redundant diverse software implementations in one hardware channel. Using different hardware resources (e.g. different RAM, ROM memory ranges) can increase the diagnostic coverage. For more details see ISO26262-5 D.2.3.4 technique. [end]

Suggested: [covers: SW_PROG_FLOW_MONIT] this safety mechanism checks the sequence of executed program tasks in order to detect a defective program sequence. A defective program sequence exists if the individual tasks of a program (e.g. software modules, functions or statements) are processed in the wrong sequence. For more details see ISO26262-5 D.2.9.5 technique. These specific software countermeasures can run once per FTTI. [end]

Suggested: [covers: SW_PROG_TEMPORAL_MONIT] this safety mechanism is intended to supervise the reliability of program execution in consideration of periodicity and maximum timing constraints of periodicity. For more details see ISO26262-5 D.2.9.5 technique. These specific software countermeasures can run once per FTTI. [end]

Suggested: [covers: SW_CORE_SELF_TEST] specific software countermeasures are implemented to detect Core permanent faults. These specific software countermeasures can run once after the Power-On Reset (POR) before the safety function starts and/or once per FTTI depending on the type of faults to be detected. [end]

2.3 System Clock and Frequency-Modulated Phase-Locked Loop (FMPLL)

External oscillator (XTAL_OSC) and FMPLL output are monitored by the hardware module called Clock Quality Monitor (CQM).

Suggested: [covers: HW_CQM] FMPLL is configured to use the external oscillator (XTAL_OSC) as their source clock and all safety relevant modules are clocked with the FMPLL generated clock signal. The CQM loss-of-clock (XTAL_OSC failure, i.e. FMPLL reference failure) and the CQM loss-of-lock (FMPLL failure) detection is enabled with relevant ISR request. The management of these errors is application-dependent. These specific software countermeasures can run once after the Power-On Reset (POR) before the safety function starts. [end]

Rationale: to reduce the impact of glitches stemming from the external quartz crystal or the RC_OSC and to check the FMPLL clock integrity.

Suggested: [covers: SW_CQM_SELF_TEST] specific software countermeasures are implemented to detect CQM permanent faults. These specific software countermeasures can run once after the Power-On Reset (POR) before the safety application starts. [end]

Implementation hint: This safety mechanism could consist of the following tests: (1) IRC frequency is measured using CQM and it is checked if the measured value is or not in the correct range; (2) CQM Crystal clock monitor feature is checked setting a wrong configuration and servicing the CQM ISR which occurs in this case; (3) CQM0 PLL clock monitor feature is checked setting a wrong configuration and CGM.CQM_0_LFREFR_A and servicing the CQM ISR which occurs in this case.

Suggested: [covers: SW_PLL_SELF_TEST] SW checks that the device is using FMPLL clock as system clock and the integrity of the CQM module before the safety function starts. [end]

Rationale: to check the correctness of FMPLL configuration and the integrity of the CQM module.

Implementation hint: e.g. a wrong PLL configuration is set in order to inject a loss-of-lock and then the CQM response is tested.

2.4 General-Purpose Static RAM (SRAM)

Suggested: [covers: HW_SRAM_ECC] the system SRAM is protected by a single error correction/dual error detection (SEC/DED) ECC scheme. The SEC/DED ECC scheme reporting is configured (interrupt request). The SRAM SEC/DED concerns data and not the addresses. Such configuration has done once after the Power-On Reset (POR) before the safety application starts. [end]

Suggested: [covers: SW_SRAM_SELF_TEST] in order to increase the diagnostic coverage, specific software countermeasures are implemented to detect RAM address logic faults. These specific software countermeasures can run once per FTTI. [end]

Rationale: to verify the integrity of RAM address logic.

Suggested: [covers: SW_SRAM_ECC_SELF_TEST] in order to increase the diagnostic coverage, specific software countermeasures are implemented to detect fault in the RAM ECC logic. The aim is to assure that correct data are not accidentally modified and that bit errors are properly corrected/detected. These specific software countermeasures can run once per FTTI. [end]

Rationale: to check the correct working of RAM ECC logic.

Implementation hint: ECSM module can force the generation of single-bit and/or double-bit data inversions in RAM allowing the check of the ECC logic. In particular ECSM module can generate errors during data write cycles, such that subsequent reads of the corrupted

address locations generate ECC events, either single-bit corrections or double-bit non-correctable errors that are terminated with an error response.

Suggested: [covers: SW_SRAM_MBIST] in order to increase the diagnostic coverage, one or more industry-standard MBIST algorithms (such as the "March" algorithm, the checkerboard algorithm, the varied pattern background algorithm and the array BIST) are implemented by software to protect the system SRAM against hardware dormant faults. The implemented MBIST algorithms can run once after the Power-On Reset (POR). [end]

Rationale: to check the integrity of the RAM memory.

2.5 FLASH memory

Suggested: [covers: HW_FLASH_ECC] FLASH memory is protected by a single error correction/dual error detection (SEC/DED) ECC scheme. The SEC/DED ECC scheme reporting is configured (interrupt request). The FLASH memory SEC/DED concerns data and not the addresses. Such configuration is done once after the Power-On Reset (POR) before the safety function starts. [end]

Suggested: [covers: SW_FLASH_SELF_TEST] in order to increase the diagnostic coverage, specific software countermeasures are implemented to detect FLASH memory address logic faults. These specific software countermeasures can run once per FTTI. [end]

Rationale: to check the correct working of FLASH memory address logic.

Implementation hint: e.g. known pattern can be read.

Suggested: [covers: SW_FLASH_ECC_SELF_TEST] in order to increase the diagnostic coverage, specific software countermeasures are implemented to detect FLASH ECC logic faults. The aim is to assure that correct data are not accidentally modified and that single bit errors are rightly corrected. These specific software countermeasures can run once per FTTI. Hardware support test called Array Integrity Self Check can be used (refer to the reference manual to have all additional details). [end]

Rationale: to verify the integrity of FLASH ECC logic.

Implementation hint: see Chapter 5: ECC logic test for further details.

Suggested: [covers: SW_FLASH_MBIST] in order to increase the diagnostic coverage, an MBIST algorithms are implemented to protect the system FLASH memory against hardware dormant faults. The implemented MBIST algorithms can run once after the Power-On Reset (POR). [end]

Rationale: to check the integrity of the FLASH memory.

Implementation hint: hardware support test called Array Integrity Self Check can be used (refer to the SPC563Mxx Reference Manual to have all additional details).

Suggested: [covers: SW_FLASH_READ_BACK] When writing flash memory, the corresponding software driver must validate the correctness of the programming of flash memory by checking relevant flash registers. Furthermore, the data that was written must be read back and then verified by software that compares it with the intended data value. [end]

Rationale: to verify that written data are coherent with the intended ones.

2.6 Interrupt Controller (INTC)

Suggested: [covers: SW_INTC_APPLICATION] integrity of the INTC module is checked. These specific software countermeasures can run once after the Power-On Reset (POR) before the safety application starts. [end]

Implementation hint: e.g. the INTC module is configured to generate some interrupt requests and the expected behavior is verified.

Suggested: [covers: SW_ISR_SELF_TEST] considering that no specific hardware protection is implemented against failures in the Interrupt Controller, spurious/missing interrupt requests caused by Electromagnetic Interference (EMI) or by bit flips in the interrupt registers of the peripherals, applications not resilient against such errors are included detection or protection software countermeasures. These specific software countermeasures can run for each interrupt request. [end]

Rationale: to verify interrupt requests are serviced correctly.

Implementation hint: e.g. spurious interrupts can be detected checking corresponding interrupt status in the interrupt status register of the related peripheral before executing the Interrupt Service Routine (ISR) code and missing interrupts, if they are synchronous, can be detected checking the program flow.

2.7 Enhanced Direct Memory Access (eDMA)

Suggested: [covers: SW_DMA_APPLICATION] considering that no specific hardware protection is implemented against failures in the eDMA, spurious/missing eDMA requests caused by Electromagnetic Interference (EMI) or by bit flips in the interrupt registers of the peripherals, applications not resilient against such errors are included detection or protection software countermeasures. These specific software countermeasures can run once after the Power-On Reset (POR) before running the SIF. [end]

Rationale: to verify eDMA requests are serviced correctly.

Implementation hint: specific software countermeasures are implemented to detect eDMA permanent faults. These specific software countermeasures are application-dependent.

2.8 Communication peripherals

The SPC563Mxx includes the following communication peripherals:

- FlexCAN
- Deserial Serial Peripheral Interface (DSPI)
- Enhanced Serial Communication Interface (eSCI)

Suggested: [covers: SW_xxx_FT_COMM_PROT] an appropriate safety software protocol should be implemented (e.g. Fault Tolerant Communication Layer or FTCOM) for any communication peripheral employed to meet safety application requirements. [end]

2.9 I/O peripherals

The SPC563Mxx device includes the following I/O peripherals:

- System Integration Unit Lite (SIUL)
- Configurable Enhanced Modular IO Subsystem (eMIOS200)
- Enhanced Time Processing Unit (eTPU2)

With the assumption that such modules are used to read/write digital input/output or PWM, the measures described in the following sections are used to protect their integrity.

2.9.1 Read digital inputs

Suggested: [covers: SW_READ_DIGITAL_INPUT] safety relevant digital inputs are acquired redundantly. Each double acquisition can be implemented using two pads configured as GPIs by the SIU unit. Digital input signal are applied on selected pads in order to be acquired and then the acquired values are compared to each other by software. [end]

Rationale: to verify that the two input values match.

Implementation hint: plausibility check on a single acquisition may replace the redundant acquisition. This hint is a special case of deviating from recommended requirements as described in the Preface. The two selected pads are not be physically adjacent to minimize CCFs. Each pad not dedicated to a specific function can be configured as GPIO with the exception of ADC pads, as they can only be configured as GPIs. SIU pads can be configured via the relevant pad configuration registers (PCRn).

2.9.2 Read PWM inputs

Suggested: [covers: SW_READ_PWM_INPUT] safety relevant digital inputs are acquired redundantly. Each double acquisition can be implemented using two pads configured as eMIOS200 channel by the SIU unit and configured with input capture feature by eMIOS200 module. PWM input signal is applied on selected pads in order to be acquired (duty cycle and period) and compared by software. [end]

Rationale: to verify that the two sets of data match.

Implementation hint: the comparison must take into account possible approximation because of different capturing of the input asynchronous signals. Each pad not dedicated to a specific function can be configured as GPIO with the exception of ADC pads, as they can only be configured as GPIs. SIU pads can be configured via the relevant pad configuration registers (PCRn). The two selected pads are not be physically adjacent to minimize CCFs. PWM input signal can be generated using one other eMIOS200 channel correctly configured or one eTPU2 channel correctly configured and the eTPU2 channels instead of the eMIOS200 channels can be used.

2.9.3 Write digital outputs

Suggested: [covers: SW_WRT_DIGITAL_OUTPUT] safety relevant digital outputs are written either redundantly or with read-back. Write digital output operation can be implemented as single write digital output with read-back or double write digital output. [end]

Rationale: to verify that the two output values match.

Implementation hint: plausibility check on a single acquisition may replace the redundant write or the write with read-back. This hint is a special case of deviating from recommended

requirements as described in the Preface. Each pad not dedicated to a specific function can be configured as GPIO with the exception of ADC pads, as they can only be configured as GPIs. SIU pads can be configured via the relevant pad configuration registers (PCRn).

Single write digital output with read-back

Implementation hint: SIU pads are used to perform a single write digital output with read-back. The read-back is done using the external configuration or the internal. SIU pads are configured as follows (see also *Figure 1*):

- External read-back: one SIU pad is configured to allow read-back of the output write on the selected SIU pad and the loop-back is done with an external connection outside the device. Using this configuration, external pins are used. In case of External read-back, the two selected pads are not be physically adjacent;
- Internal read-back[(c)]: one SIU pad is configured to allow read-back of the output write on the selected SIU pad via an internal read path. Using this configuration, a single external pin is used.

Double Write Digital Outputs

Implementation hint: SIU pads are used to perform a double write digital output. SIU pads are correctly configured and the output write of the selected channels are implemented following these guidelines:

- The two outputs are written with a single instruction to the appropriate register;
- The output register is read-back[(d)].

The two selected pads are not be physically adjacent to minimize CCFs. Each pad not dedicated to a specific function can be configured as GPIO with the exception of ADC pads, as they can only be configured as GPIs. SIU pads can be configured via the relevant pad configuration registers (PCRn).To write two (or more) GPIOs with a single write instruction, the Parallel GPIO Pad Data Out (PGPDOx) register can be used. User has to take care that the two selected GPIOs are controlled by the same PGPDOx register. To protect the value of the other GPIOs that belong to the same PGPDOx register, the Masked Parallel GPIO Pad Data Out (MPGPDOx) register is properly configured before writing the PGPDOx register.

2.9.4 Write PWM outputs

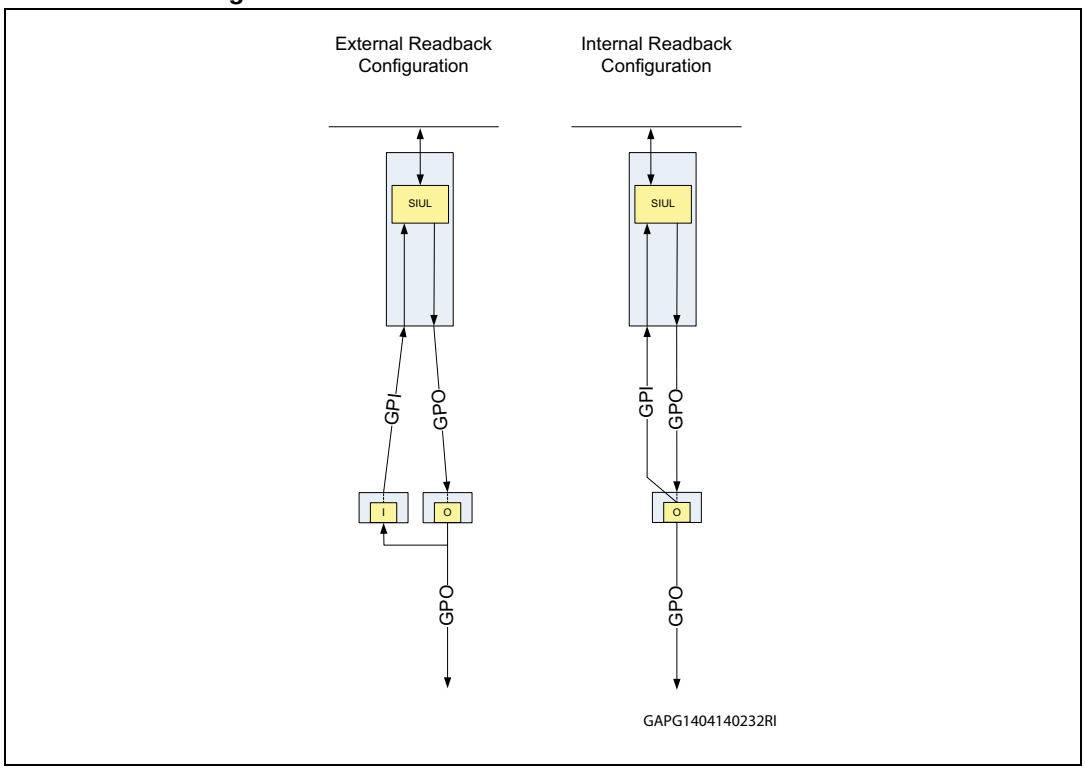
Suggested: [covers: SW_WRT_PWM_OUTPUT] some PWM outputs are written either redundantly or with read-back. Write PWM output operation can be implemented as single write PWM output with read-back or double write PWM output. [end]

Rationale: to verify that the two sets of data match.

c. Internal read back does not cover package faults (e.g. wire bond). Refer to the specific reference manual to verify the availability of the internal read path.

d. This is not strictly requested. The SPC563Mxx can demand to an external device to check the integrity of the PWM generation.

Figure 1. Block scheme of external/internal read-back



Single write PWM output with read-back

Implementation hint: Two pins are used to implement this function. One pin to generate the PWM output and another pin to read-back this PWM to check its integrity. Each single write with read-back can be implemented using pads configured as eMIOS200 channel by the SIU unit and configured as PWM by eMIOS200 module. PWM set of data (duty cycle and period) is applied by software. The read-back is done using the external configuration or the internal. SIU pads is configured as follows:

- External read-back: one SIU pad is configured as eMIOS200 channel by the SIU unit and configured with input capture feature by eMIOS200 module to allow read-back of the output written on the output pad. The loop-back is done with an external connection outside the device. In case of External read-back, the two selected pads are not physically adjacent;
- Internal read-back[(e)]: one SIU pad is configured as eMIOS200 channel by the SIU unit and configured with input capture feature by eMIOS200 module to allow read-back of the output write on the selected pad. The loop-back is done via an internal read path.

The two selected pads are not physically adjacent to minimize CCFs. SIU pads can be configured via the relevant pad configuration registers (PCRn). eTPU2 channels instead of eMIOS200 channels can be used.

Double Write PWM Outputs

Implementation hint: Each double write can be implemented using two pads configured as eMIOS200 channel by the SIU unit and configured as PWM by eMIOS200 module. PWM

e. Internal read back does not cover package faults (e.g. wire bond). Refer to the specific reference manual to verify the availability of the internal read path.

set of data (duty cycle and period) are applied by software. The two selected pads are not physically adjacent to minimize CCFs. SIU pads can be configured via the relevant pad configuration registers (PCRn). eTPU2 channels instead of eMIOS200 channels can be used.

2.9.5 Enhanced Time Processing Unit (eTPU2)

Suggested: [covers: SW_ETPU_APPLICATION] considering that no specific hardware protection is implemented against failures in the eTPU2, spurious/missing eTPU2 actions caused by Electromagnetic Interference (EMI) or by bit flips in the interrupt registers of the peripherals, applications not resilient against such errors include detection or protection software countermeasures. These specific software countermeasures can run once after the Power-On Reset (POR) before the safety function starts and once per FTTI. [end]

Rationale: to verify eTPU2 requests are serviced correctly.

Implementation hint: specific software countermeasures are implemented to detect eTPU2 permanent faults. These specific software countermeasures are application-dependent.

2.10 Enhanced Queued Analog-to-Digital Converter (eQADC)

The SPC563Mxx device is equipped with one eQADC module integrating two analog-to-digital converter macro-cells.

Suggested: [covers: SW_ADC_SELF_TEST] acquisition of some reference voltages is done. These specific software countermeasures can run once after the Power-On Reset (POR) before the safety function starts and/or once per FTTI. [end]

Rationale: to check the integrity of the eQADC module.

Implementation hint: some eQADC channels are internally connected to some reference voltages as Buffered Band Gap, Reference Voltage for 1.2 V LVD and so on (refer to the SPC563Mxx Reference Manual to have all additional details).

Moreover recommendation is to acquire analog input redundantly.

This module, if safety relevant, is used to implement the following function:

- Double read analog inputs;
- Additional mechanisms.

2.10.1 Double read analog inputs

Suggested: [covers: SW_DOUBLE_READ_AN_INPUT] safety relevant analog inputs are acquired redundantly using both analog-to-digital converter macro-cells integrated in the eQADC module. The measured values are compared by software. [end]

Rationale: to verify that the two measured analog input values match.

Implementation hint: shared channels are not be used for double read operation in order to avoid CCFs due to the pad sharing. SIU pads can be configured via the relevant pad configuration registers (PCRn).

2.10.2 Additional mechanisms

The two analog-to-digital converter macro-cells share the same digital interface. To increase the diagnostic coverage against failures impacting this common logic, some additional counter measures can be developed. For example:

- Oversampling;
- Plausibility check.

Suggested: [covers: SW_OVERSAMPLING_AN_INPUT] the analog inputs are acquired redundantly in time. [end]

Rationale: to increase the diagnostic coverage.

Implementation hint: the sampling rate is significantly higher than the Nyquist Frequency related to the input signal. The acquired values are compared by software in order to verify the correlation. In case of fault, the acquired values are not correlated with themselves. Against random faults, at least three consecutive analog values are acquired for each analog input.

2.11 Temperature sensor

SPC563Mxx devices are equipped with a temperature sensor in order to monitor the device temperature. This temperature sensor generates a voltage that increases linearly with temperature and that can be read by software using the on-board eQADC module, so the read value can be used with the band-gap voltage and constants stored in flash memory during factory test to calculate device junction temperature.

Suggested: the temperature sensor output voltage is read by software and the corresponding temperature is compared with the upper limit of the operating range. In case an over-temperature fault is detected, the device is set to a safe state. This check runs once per FTTI.

Rationale: to detect over-temperature faults.

Implementation hint: to set the proper operating range threshold, the temperature sensor accuracy of 10° C and the maximum operating junction temperature of 150 °C (see device data sheet,*Chapter Appendix B: Document references*) are considered.

Note: External temperature sensor could be used to check internal temperature sensor output.

2.12 Software Watchdog Timer (SWT)

Suggested: [covers: HW_INT_SWT] SWT module is used to implement control flow monitoring function. The SWT is clocked by oscillator clock. These specific software countermeasures can run once after the Power-On Reset (POR) before the safety function starts. [end] However, other control flow monitoring approaches that do not use the SWT may also be used. SPC563Mxx devices provide the hardware support (SWT) to implement both control flow monitoring and temporal flow monitoring methods.

Rationale: to detect a defective program sequence.

Implementation hint: SWT can be enabled asserting the bit SWT_MCR[WEN] and the configuration registers can be hard-locked asserting the bit SWT_MCR[HLK]. The timeout register (SWT_TO) must contain a 32-bit value that represents a timeout less than the FTTI. Before the safety function is executed, software must verify that the SWT is enabled

checking the bit SWT_MCR[WEN]. If Windowed mode and Keyed Service mode (two pseudo-random key values used to service the watchdog) are enabled, it is possible to reach a high effective temporal flow monitoring.

2.13 Multi-Layer AHB Crossbar Switch (XBAR)

Suggested: [covers: SW_AXBS_SELF_TEST] the configuration and the integrity of the XBAR are checked. These specific software countermeasures can run once after the Power-On Reset (POR) before the safety function starts and/or once per FTTI. [end]

Implementation hint: e.g. the integrity of the XBAR module can be checked reading a checking pattern (stored in the FLASH memory) with the master Core and eDMA, calculating the CRC of the checking pattern and comparing this with the expected one. Different checking patterns (stored in different location of the FLASH memory) could be chosen for each FTTI.

2.14 Peripheral Bridge (PBRIDGE)

Suggested: [covers: SW_AIPS_SELF_TEST] PBRIDGE is configured in order to ensure that all bus masters (Core, eDMA and FlexRay) can access only their allocated resources according to their access rights. The configuration and the correct working of the PBRIDGE are checked. These specific software countermeasures can run once after the Power-On Reset (POR) before the safety function starts and/or once per FTTI. [end]

Rationale: to avoid to give access to the device resources to unauthorized master and to deny access to authorized master.

Implementation hint: e.g. the integrity of the PBRIDGE module can be checked calculating the CRC of the configuration registers value of 3 IPs and comparing each one with the expected one. Different IPs could be chosen for each FTTI.

2.15 Power Management Controller (PMC)

SPC563Mxx devices use three supply voltages, nominally 5V, 3.3V and 1.2V. The 5V supply voltage must be supplied from the outside while the other supply voltages are supplied by internal regulators. Moreover, SPC563Mxx devices embed LVI for all supply voltages. The PMC controls the internal regulators and the LVI circuits.

Suggested: [covers: HW_LVI] LVI failure reaction for all supply voltages is configured (system reset or interrupt request). These specific software countermeasures can run once after the Power-On Reset (POR) before the safety function starts. [end]

Rationale: to check if supply voltages are in the correct operation range.

Suggested: [covers: SW_LVI_SELF_TEST] LVI circuits operation (for supply voltages generated by internal regulators, i.e. 3.3V and 1.2V) is checked. These specific software countermeasures can run once after the Power-On Reset (POR) before running the SIF. [end]

Implementation hint: the output of each internal regulator can be set to a value lower than the LVI threshold value configuring the PMC_ TRIMR register. According to this, enabling only the interrupt request as LVI failure reaction, the generation of the LVI interrupt requests

confirms the correctness of LVI circuits operation. Then, the correct value of the PMC_TRIMR register can be restored.

Suggested: [covers: SW_POR_SELF_TEST] correct execution of Power-On Reset sequence is checked. These specific software countermeasures can run once after the Power-On Reset (POR) before the safety function starts. [end]

Implementation hint: e.g. reserved RAM is used to store a key which can be used if the current reset is a POR or not according to the POR bit in the ECSM_MRSR register. Moreover the default reset value of the registers of each IP can be checked.

2.16 Error Correction Status Module (ECSM)

The ECSM reports ECC/EDC failures affecting both volatile and non-volatile memories. The ECSM reports detection/correction of single-bit errors and detection of double-bit ones. ECC functionality concerns data and not the addresses. ECC is automatically calculated on memory write accesses and is checked while read accesses are executed on memory.

Suggested: to enable ECC reporting logic in the ECSM in order to provide an optional managing failures interrupt mechanism. In addition to the interrupt generation, the ECSM reports specific information about the failure (memory address, attributes and data, bus master number, etc.) which can be useful for subsequent management of the failure.

Rationale: to manage failures.

Suggested: [covers: SW_ECSM_SELF_TEST] specific software countermeasures are implemented to detect ECSM permanent faults. These specific software countermeasures can run once after the Power-On Reset (POR) before the safety function starts. [end]

2.17 Periodic Interrupt Timer (PIT_RTI)

Suggested: [covers: SW_PIT_APPLICATION] Specific software countermeasures are implemented to detect PIT transient and permanent faults. These specific software countermeasures can run once after the Power-On Reset (POR) before the safety function starts and/or once per FTTI. [end]

Rationale: to detect possible PIT failures.

Implementation hint: Test implementation is application-dependent. In general PIT module is used in such a way that a possible failure is detected by the SWT module (or other means).

2.18 System Timer Module (STM)

Suggested: [covers: SW_STM_APPLICATION] Specific software countermeasures is implemented to detect STM transient and permanent faults. These specific software countermeasures can run once after the Power-On Reset (POR) before running the SIF and/or once per FTTI. [end]

Rationale: to detect possible STM failures.

Implementation hint: Test implementation is application-dependent. In general STM is used in such a way that a possible failure is detected by the SWT module (or other means).

2.19 All safety relevant modules

Suggested: [covers: SW_CRC_CONF_REG] CRC signature iss used to check the correctness of the content of the configuration registers of each safety-related module. This check runs once per FTTI. [end]

Implementation hint: E.g. the CRC signature of the content of the configuration registers of each safety-related module is calculated off-line. At run time, the same CRC signature is calculated by software within the FTTI. The run-time calculated CRC signature is then compared to the expected one, i.e. the off line calculated CRC signature. CRC signature could be calculated by software using one or more industry standard CRC algorithms. To avoid CPU overloading, the EDMA module can be used to support the data transfer from the registers under check to the RAM.

3 Functions of external devices for safety applications

This section gives an overview of the external components suggested to use with the SPC563MXX device.

Suggested: at system level some countermeasures have to be placed in order to bring the safety-critical outputs to their safe state (e.g., by pull-up or pull-down resistors).

It should be noted that the failure rates of external components are not included in FMEDA of the SPC563Mxx device and have to be included in the system FMEDA by the user himself.

3.1 External Watchdog function (EXWD)

Suggested: [covers: SW_EXT_SWT] an external low-cost device, acting as system supervisor, provides also a watchdog to cover complete failures of the SPC563Mxx device for safety applications. It is triggered periodically by the SPC563Mxx device. [end]

Rationale: to detect CCF as a complete failure of the device.

Some common causes of failure (e.g., failure on power supply) are detected because the software no longer triggers the watchdog.

If a failure is detected, the EXWD moves, and maintains, the system (ECU level) to a Safe state condition within the FTTI (e.g., the EXWD disconnects from the power supply the SPC563Mxx device).

The user can choose how to implement the watchdog communication between the SPC563Mxx device and the external device (for example, communication via serial link or via toggling pin).

3.2 Power Supply Monitor function (PSM)

The SPC563Mxx device embeds LVI for all internal supplies. Latent failures impacting these LVIs can’t be detected.

Suggested: [covers: SW_EXT_SUPPLY_MON] an external low-cost device, acting as system supervisor, is provided also over-/under-voltage monitor for the SPC563Mxx on all supplies available externally. [end]

Rationale: to ensure voltage power supply is within the defined operating range.

If the voltage power supply is out of the defined operating range, the PSM moves, and maintains, the system (ECU level) to a Safe state condition within the FTTI (e.g., the PSM disconnects from the power supply the SPC563Mxx device).

For the voltage power supply operating range, please refer to the SPC563MXX device data sheet.

It should be noted that an over voltage outside the specified range may cause permanent damage to the SPC563Mxx device even if kept in reset.

3.3 PWM Output Monitor function (PWMM)

The eMIOS200 module and the eTPU2 module integrated in the SPC563Mxx device can generate PWM output signals.

In general, if the safety application uses these PWM output signals to control an actuator with short safety time against wrong control (such as the inverter of a three-phase motor control application with a dead-time requirements to avoid short circuits destroying the inverter and the motor), those requirements are supervised externally if the failure reaction delay within the SPC563Mxx device can exceed the safety time of the actuator.

The distinctive features that should be managed by the external device are the correctness of inserted dead-time and the occurrence of an open-circuit and/or short-circuit to supply or ground.

Suggested: [covers: SW_EXT_PWM_MON] an external low-cost device, acting as system supervisor, provides also a PWM monitor to check the generated PWM output signals.

Rationale: to check the accuracy of the PWM output signals.

If a failure is detected, the PWMM moves, and maintains, the system (ECU level) to a Safe state condition within the FTTI (e.g., the PWMM disconnects from the power supply the SPC563Mxx device).

Implementation hint: in case PWM signals drive the switches of a power stage, eMIOS200 channels or eTPU2 channels cannot be used to detect a dead-time fault because its failure indication time is normally greater than the time enough to produce a physical permanent failure of the power stage.

4 ECC logic test

4.1 Overview

This section describes the required information on how to develop the software for such ECC logic test.

The goal is to ensure high coverage of the ECC logic faults with minimum performance penalty to customer's application. Thus, the performance penalty must be less than 2% (e.g. the test time should be less than 200 µs considering a FTTI of 10 ms).

The SPC563MXX FLASH memory has a UTEST (user-test) mode ECC logic check feature which can be utilized for this ECC logic test. A data pattern with walking 0 through data and ECC parity bits can be applied during the ECC logic check procedure to achieve high fault coverage of the ECC logic and fast execution.

4.2 Data pattern – Walking 0

To reach the needed performances the use of the data pattern with walking 0 through data and ECC parity bits must be used. Table 1 shows the data pattern.

Table 1. Data pattern used by the ECC logic test

Data vector number	8-bit ECC parity bits	64-bit data bits
0	0xFF	0xFFFF_FFFF_FFFF_FFFE
1	0xFF	0xFFFF_FFFF_FFFF_FFFD
2	0xFF	0xFFFF_FFFF_FFFF_FFFB
3	0xFF	0xFFFF_FFFF_FFFF_FFF7
4	0xFF	0xFFFF_FFFF_FFFF_FFEF
5	0xFF	0xFFFF_FFFF_FFFF_FFDF
6	0xFF	0xFFFF_FFFF_FFFF_FFBF
7	0xFF	0xFFFF_FFFF_FFFF_FF7F
…	…	…
62	0xFF	0xBFFF_FFFF_FFFF_FFFF
63	0xFF	0x7FFF_FFFF_FFFF_FFFF
64	0xFE	0xFFFF_FFFF_FFFF_FFFF
65	0xFD	0xFFFF_FFFF_FFFF_FFFF
…	…	…
71	0x7F	0xFFFF_FFFF_FFFF_FFFF
72	0xFF	0xFFFF_FFFF_FFFF_FFFF

It is important to note that for double word data = 0xFFFF_FFFF_FFFF_FFFF, the correct ECC check bits should be 0xFF. Therefore, every data vector in the data pattern in *Table 1: Data pattern used by the ECC logic test*, except the last one, contains a single-bit ECC error and results in a single-bit correction.

4.3 UTEST mode ECC logic check

The procedure to use the UTEST mode ECC logic check is listed as below:

1. Enable UTEST mode (Write 0xF9F9_9999 to UT0 register, UT0[UTE] is set).
2. Write UT0[SBCE] to 1 (to enable single-bit error correction visibility).
3. Write UT0[EIE] to 1.
4. Write UT0[DSI], UT1[DAI] and/or UT2[DAI] bits to provide data and check bit values to be read. Single or Double bit detections/corrections can be simulated by properly choosing Data and Check Bit combinations.
5. Write double word address to receive the data inputted in step 3 into the ADR register.
6. Reads can now be done through the BIU in a Read Request type fashion. In the event of a BIU read requested from an address that matches the address in the ADR register, expected data, and corrections or detections should be observed based on data written into the UT0[DSI], UT1[DAI] and/or UT2[DAI] registers. MCR[EER] and MCR[SBCSBC] can be checked to evaluate the status of reads done.
7. Repeat steps 4 to 6 for all the data vectors in the proposed test data pattern.
8. Once completed, clear the UT0[EIE] bit to 0.

4.4 Fault coverage and execution time

The described ECC logic test reaches around 90% of fault coverage of ECC decode logic.

The execution of the test code takes about 200 µs at 80 MHz, room temperature and nominal voltages.

Appendix A Further information

A.1 Conventions and terminology

Table 2 hows the list of conventions for this document.

Table 2. List of conventions and terminology

Convention	Description
Error	Discrepancy between a computed, observed, or measured value or condition and the true, specified or theoretically correct value or condition.
Fault	Abnormal condition that may cause a reduction in, or loss of, the capability of a functional unit to perform a required function.
Failure	The termination of the ability of a functional unit to perform a required function.

A.2 Acronyms and abbreviations

A short list of acronyms and abbreviations used in this document is reported in *Table 3*.

Table 3. Acronyms and abbreviations

Term	Meaning
CCF	Common Cause Failure
CRC	Cyclic Redundancy Check
DED	Dual Error Detection
ECC	Error Correcting Code
ECSM	Error Correction Status Module
eDMA	Enhanced Direct Memory Access
EXWD	External Watchdog function
eQADC	Enhanced Queued Analog-to-Digital Converter
FMEDA	Failure Modes, Effects and Diagnostic Analysis
FMPLL	Frequency-Modulated Phase-Locked Loop
FTTI	Fault Tolerant Time Interval
GPIO	General Purpose Input/Output
LBIST	Logic Built-In Self-Test
LVI	Low Voltage Inhibit
MBIST	Memory Built-In Self-Test
MCU	Microcontroller Unit
PMC	Power Management Controller
PSM	Power Supply Monitor function
PWM	Pulse Width Modulation

Table 3. Acronyms and abbreviations (continued)

Term	Meaning
SEC	Single Error Correction
SWT	Software Watchdog Timer

Appendix B Document references

- *32-bit Power Architecture® based MCU for automotive powertrain applications* (SPC563M64L5, SPC563M64L7, DocID 14642)
- *SPC563Mxx - 32-bit Power Architecture® based MCU with up to 1.5 Mbyte Flash and 111 Kbyte RAM memories* (Reference manual, RM0015 Doc ID 14499)

Revision history

Table 4. Document revision history

Date	Revision	Changes
05-May-2014	1	Initial release