MICROCHIP

Sample

MCU Safety Manual

Introduction

WARNING This document is for demonstration purposes only: It describes a non-existing product, and important information is intentionally left out. This document shall not be used as a reference for developing safety applications. The safety manuals for the specific products shall be used when designing safety critical applications – these can be obtained from Microchip Direct.

This manual discusses the requirements for the use of the Microchip MCU devices as a Safety Element out of Context (SEooC) in functional safety-relevant applications and provides guidelines for the proper use of Microchip MCU devices in ISO 26262 ASIL B and IEC 61508 SIL 2 rated applications. It will guide the System Integrator with the necessary steps to integrate the product into their application.

The System Integrator shall fulfill all Assumptions of Use (AoU) listed in 7. System Integrator Responsibilities, and for the selected diagnostics, all Hardware Safety Requirements (HWSR) and Software Safety Requirements (SWSR) are listed in 5. Hardware Requirements on System Level and 6. Software Requirements on System Level, respectively. The AoUs, HWSR, and SWSR were assumed to be fulfilled when analyzing the failure rates and Fault metrics in the provided FMEDA. The AoUs are summarized in section 11. Assumptions of Use (AoU).

If one or more AoUs, HWSR, or SWSR are not fulfilled, the System Integrators shall show that an alternative solution is similarly efficient concerning the safety requirement in question or show that the particular issue is irrelevant for their application (for example, the module is not used).

If these alternatives are not possible, the System Integrators shall estimate how much the failure rate increases and the failure metrics decrease due to the deviation. Otherwise, the FMEDA provided with Microchip devices is not valid.

This Safety Manual refers to the International Standard ISO 26262, 2nd Edition 2018 and to the International Standard IEC 61508, Edition 2.0 2010-04.

It may refer to MCU products that are not available in an automotive version since the manual is also used for non-automotive safety applications.

Note: Although a higher system-level ASIL/SIL rating can be achieved with this product, this manual's primary focus is on ISO 26262 ASIL B and IEC 61508 SIL 2 compliance. Diagnostic methods described within and coverage calculations obtained from the FMEDA can be used as a guide to determine additional compliance levels.

Table of Contents

1. Abbreviations and Definitions

Table 1-1. Abbreviations

AoU	Assumptions of Use
AoU-FMEDA	Assumptions of Use: Failure Modes, Effects and Diagnostic Analysis
AoU-GEN	Assumptions of Use: General
AoU-SIR	Assumptions of Use: System Integrator Responsibility
AoU-SLD_HW	Assumptions of Use: System Level Design – Hardware
AoU-SLD_SW	Assumptions of Use: System Level Design – Software
ASIL	Automotive Safety Integrity Level
SiP	System-in-Package
SBC	System Basis Chip
CAN	Controller Area Network
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
DFA	Dependent Failure Analysis
DMA	Direct Memory Access controller
DTI	Diagnostic Test Interval
E2E	End-to-End Communication Protection
EEPROM	Electrically Erasable Programmable Read-Only Memory
E/E/PS	Electric/Electronic/Programmable
EUC	Equipment Under Control
FDTI	Fault Detection Time Interval
FHTI	Fault Handling Time Interval
FIT	Failure In Time
FMEA	Failure Mode Effects Analysis
FMEDA	Failure Modes, Effects and Diagnostic Analysis
FRTI	Fault Reaction Time Interval
FTTI	Fault Tolerant Time Interval
ISR	Interrupt Service Routine
LIN	Local Interconnect Network
MCU	Microcontroller Unit
MTBF	Mean Time Between Failures; inverse of FIT Rate
NMI	Non-Maskable Interrupt
PCN	Product Change Notice
PPAP	Production Part Approval Process
PST	Process Safety Time

RAM	Random Access Memory
RPN	Risk Priority Number
SEooC	Safety Element out of Context
SFF	Safe Failure Fraction
SIL	Safety Integrity Level (IEC 61508)
SRAM	Static Random Access Memory
TLSR	Top-Level Safety Requirement
TSR	Technical Safety Requirement

Table 1-2. ISO 26262, Second Edition 2018 – Safety Basic Notations

Automotive Safety Integrity Level (ASIL)	It is a level (one out of four: ‘A’ being the least stringent, ‘D’ being the most stringent) used to define the item’s or element’s ISO 26262 requirements and Safety measures to avoid unreasonable risk.
AoU	Assumptions on the conditions of the semiconductor component usage. All AoU shall be verified and fulfilled by the System Integrator of the semiconductor component.
FIT Rate	The frequency with which something fails expressed in Failures In Time (FIT); one FIT equals one failure per one billion hours of device operation
Functional Safety	It is the absence of unreasonable risk due to hazards caused by the malfunctioning behavior of a system
Harm	It is the physical injury or damage to the health of persons
Hazard	It is a potential source of harm caused by a malfunctioning behavior of an item
Hazard Analysis and Risk Assessment (HARA)	It is a methodology for the identification and the categorization of hazardous events, and the specification of Safety Goals and ASILs, targeted to the prevention or mitigation of the associated hazards in order to avoid an unreasonable risk.
Residual FIT Rate	The FIT rate adjusted based on actual circuitry usage and failure diagnostics coverage
Risk	A combination of the probability of occurrence of harm and the severity of that harm. It is a measure of the likelihood and severity of an event resulting in loss or injury.
Safe State	An operating mode without an unreasonable level of risk. Upon detection of failure, the Safe State is the state in which a device enters to minimize any harm related to the failure.
Safety Goal	The safety requirement as a result of the hazard analysis and risk assessment. Generally applies to the item level.
Safety Mechanism	Technical solution implemented by Electrical and/or Electronic (E/E) functions, or elements targeted to detect Faults or control failures to achieve and/or maintain a safe operational state
SEooC	Safety Element out of Context. It refers to a safety-related element which is not developed in the context of a specific item.
System	Set of components that relates at least a sensor, a controller and an actuator with one another

System Integrator	The person who is responsible for integrating the SEooC into the system
Top-Level Safety Requirement	An assumed system-level safety requirement considered during the definition of lower-level hardware and/or software safety requirements required to build the device safety concept. Generally applies to semiconductor components and software parts.
V-Model	A project methodology which follows a hierarchical design approach. It starts with a high-level design, followed by detailed design, followed by testing of the detailed design and then testing of the higher level design.

Table 1-3. ISO 26262, Second Edition 2018 – Faults and Failures: Definitions

Cascading Failure	It is an element Failure resulting from an internal or external (to the element) root cause causing a Failure of one or more elements
Common Cause Failure (CCF)	Failure of two or more elements of an item resulting from a single specific event or root cause. This is a Random Failure Mode in which two or more components fail due to the same reason. Unlike a Systematic Failure, predictions of Common Cause Failures can be done only through statistical means.
Common-Mode Failure (CMF)	This is a special case of a Common Cause Failure where multiple elements fail in the same manner
Dependent Failure	Failures that have some degree of correlation to each other, so that the probability of simultaneous (two Faults occurring in two elements having the same root cause) or successive occurrence (failure of an element occurred as a consequence of the Fault/failure in another element) cannot be expressed as the product of the unconditional probabilities of each of them
Detected Fault	A Fault whose presence is detected within a prescribed time interval by a Safety Mechanism so that the Fault is not latent
Diagnostic Coverage	The proportion of the failure rate (of a hardware element) that is detected or controlled by the Safety Mechanisms implemented for that element. It represents the ability of a system to detect failures.
Dual Point Failure	It is a Failure resulting from the combination of two independent hardware Faults that leads to the violation of a Safety Goal
Error	It is the discrepancy between a computed, observed or measured value or condition, and the true, specified or theoretically correct value or condition
Failure	When an element of a system stops performing the action/function for which it was designed
Failure Mode	The manner in which a device (element or item) fails. Failure Modes can be broadly categorized as Safe Detected (SD), Dangerous Detected (DD), Safe Undetected (SU) and Dangerous Undetected (DU).
Failure Rate	It is the probable density of Failure divided by the probability of survival for a hardware element
Fault	An abnormal operating condition that causes an element of a system to fail
Fault Tolerance	It is the ability to deliver a functionality in the presence of one or more Faults
Independent Failures	They are the Failures whose probability of simultaneous or successive occurrence can be expressed as the simple product of their unconditional probabilities

Latent Fault	A Multiple Point Fault whose presence is neither detected by a Safety Mechanism nor perceived by the driver within the Multiple Point Fault detection interval. It is essentially a Multiple Point Fault that is present in the system, but hidden from detection. A Latent Fault does not, by itself, result in unsafe operation; however, a Latent Fault can result in a failure to detect an unsafe condition. This can occur, for example, when the detection circuitry itself fails or is faulty.
Multiple Point Fault/Failure	The combination of an individual Fault with other independent Faults that leads to the violation of a Safety Goal. Dual point failures are special cases of Multiple Point Failures: In such cases, one Fault affects a safety-related element and the second Fault affects the corresponding Safety Mechanism put in place to address the first Fault.
Random Hardware Failure	A failure that can occur unpredictably during the lifetime of a hardware element; the Random Hardware Failure follows a probability distribution. These failures occur at random and have many causes. Some of the more common ones are manufacturing process defects in the device, noise injected into the system, data corruption, energy surges that damage the device or crack-inducing mechanical stress. Random failures can be statistically predicted and are used to establish the probability of failure. Random failures can result in permanent/hard or recoverable/soft errors. Hard failures cause permanent damage to the component, where the system is unable to continue normal operation. Without compensation for the damage, the system has to be placed into a Safe State and repair is required to restate proper operation. Soft failures should be reversible through a recovery process. Soft failures can manifest as transients or steady-state conditions, which can be reset or reinitialized.
Random Hardware Fault	It is a hardware Fault characterized by a probabilistic distribution
Safe Fault	A Fault whose occurrence will not significantly increase the probability of violation of a Safety Goal. A Safe Fault can only lead to a safe failure (i.e., a system failure that does not result in a dangerous situation).
Single-Point Fault/Failure	A Fault in a HW element that is not covered by a Safety Mechanism and that leads directly to the violation of a Safety Goal. A Single-Point Failure is a failure that can occur as a result of a Single-Point Fault and can result in unsafe/dangerous operation of the system. Therefore, system diagnostics are needed to detect such Faults and assure that the system enters a safe operating state.
Systematic Failure	A failure that is related in a deterministic way to a certain cause that can only be eliminated by a change in the design or manufacturing process, operational procedures, documentation or other relevant factors. Such failures are due to deterministic non-random and predictable causes, and are not mathematically predictable. Note that basic redundancy cannot prevent Systematic Failures, as both redundant elements would have the same faulty behavior. Systematic Failures can only be eliminated by design, process or functional change. Systematic Failures can be predicted with rigorous engineering analysis and design methods.
Systematic Fault	It is a Fault whose Failure manifests in a deterministic way that can only be prevented by applying process or design measures

Table 1-4. ISO 26262, Second Edition 2018 – Fault Classification

Detected Fault	A Fault whose presence is detected within a prescribed time interval by a Safety Mechanism so that the Fault is not latent
Dual Point Fault	It is an individual Fault that, in combination with another independent Fault, leads to a Dual Point Failure
Latent Fault	A Multiple Point Fault whose presence is neither detected by a Safety Mechanism nor perceived by the driver within the Multiple Point Fault detection interval. It is essentially a Multiple Point Fault that is present in the system, but hidden from detection. A Latent Fault does not, by itself, result in unsafe operation; however, a Latent Fault can result in a failure to detect an unsafe condition. This can occur, for example, when the detection circuitry itself fails or is faulty.
Multiple Point Fault/Failure	The combination of an individual Fault with other independent Faults that leads to the violation of a Safety Goal. Dual point failures are special cases of Multiple Point Failures: in such cases, one Fault affects a safety-related element and the second Fault affects the corresponding Safety Mechanism put in place to address the first Fault.
Residual Fault	Part of a Fault (capable of violating a Safety Goal) that is not completely covered by the Fault Safety Mechanism put in place
Perceived Fault	It is a Fault that may be perceived indirectly (through deviating behavior on a vehicle level)
Single Point Fault	It is a Fault in an element that is not covered by a Safety Mechanism and that leads directly to the violation of a Safety Goal. Therefore, system Diagnostics are needed to detect such Faults and assure the system enters a Safe operating state.

Table 1-5. ISO 26262, Second Edition 2018 – Safety Measures

Compensation Method	A method that allows the system to continue normal operation once a failure has been detected
Diversity	The implementation of different solutions satisfying the same requirement with the aim of independence. It essentially involves employing different methods to achieve the required function within the system, such as using both analog and digital signals to transmit information or utilizing two unique calculations to arrive at the same answer.
Parity	Supplemental data indicating an even or odd number of '1's in a binary data stream. It is used to verify basic data integrity for data storage or transmission.
Redundancy	The existence of means, in addition to the means that would be sufficient for an element to perform a required function, or to represent information. Practically, it means having multiple elements or systems used to achieve the same function.

Table 1-6. ISO 26262, Second Edition 2018 – Redundancy Types

Functional	Parallel and diverse hardware structures or software methods applied to a single task
Informational	Extra information is included with the key data and verified for coherency, such as a parity bit, ECC, CRC checksum, etc.

Structural	Parallel and identical structures performing the same task. At the system level, this would include dual registers, memories, CPUs, controllers, etc. At the application level, this includes redundant inputs or outputs (digital or analog), redundant sensors, redundant controllers, etc.
Temporal	The same method is applied multiple times by the same hardware or software at different times (i.e., software repeats the same calculation or task at different times and compares the results)

Table 1-7. IEC 61508, Edition 2.0 2010-04 – Safety Basic Notations

Functional Safety	It is the part of the overall Safety relating to the EUC and the EUC control system that depends on the correct functioning of the Safety-related systems and on risk reduction measures
Harm	It is the physical injury or damage to the health of people, or damage to property or the environment
Hazard	It is a potential source of harm
Mode of Operation	It is the way in which a Safety function operates, which may be one of the following: • Low Demand Mode: Where the Safety function is only performed on demand in order to transfer the EUC into a specified Safe State, and where the frequency of demands is no greater than one per year • High Demand Mode: Where the Safety function is only performed on demand in order to transfer the EUC into a specified Safe State, and where the frequency of demands is greater than one per year. • Continuous Mode: Where the Safety function retains the EUC in a Safe State as part of normal operation.
Reasonably Foreseeable Misuse	It is the use of a product, process or service in a way not intended by the supplier, but which may result from readily predictable human behavior
Residual Risk	It is the risk remaining after protective measures have been taken
Risk	Risk is a combination of the probability of occurrence of harm and the severity of that harm
Safe State	It is the state of the EUC when Safety is achieved
Safety	It is the freedom from unacceptable risk
Safety Function	It is a function to be implemented by a Safety-related system or other risk reduction measures, intended to achieve or maintain a Safe State in respect of a specific hazardous event
Safety Integrity Level (SIL)	It is a discrete level (one out of a possible four) corresponding to a range of Safety integrity values, where Safety Integrity Level 4 has the highest level of Safety integrity and Safety Integrity Level 1 has the lowest
Safety Related System	It is a designated system that both • implements the required Safety Functions necessary to achieve or maintain a Safe State for the EUC • are intended to achieve (alone or with Safety-related systems and risk reduction measures) the necessary Safety integrity

Systematic Capability	It is a measure (expressed on a scale of SC 1 to SC 4) of the confidence that the systematic Safety integrity of an element meets the requirements of the specified SIL
Target Risk	It is the risk that is intended to be reached for a specific hazard, taking into account the EUC risk, together with the Safety-related systems and the other risk reduction measures
Tolerable Risk	It is a risk which is accepted in a given context based on the current values of society

Table 1-8. IEC 61508, Edition 2.0 2010-04 – Faults and Failures: Definitions

Average Frequency of a Dangerous Failure per Hour (PFH)	It is the average frequency of a Dangerous Failure of a Safety-related system to perform the specified Safety Function over a given period of time.
Average Probability of Dangerous Failure on Demand (PFDavg)	It is the mean unavailability of a Safety-related system to perform the specified Safety Function when a demand occurs
Common Cause Failure	It is a Failure, result of one or more events, causing concurrent Failures of two or more separate channels in a multiple channel system, leading to system Failure
Dangerous Failure	It is the Failure of an element/subsystem/system that plays a part in implementing the Safety Function that: a) Prevents a Safety Function from operating when required (Demand mode) or causes a Safety Function to fail (Continuous-mode) so the EUC is put into a (potentially) hazardous state, or b) Decreases the probability that the Safety Function operates correctly when required.
Dependent Failure	It is a Failure whose probability cannot be expressed as the simple product of the unconditional probabilities of the individual events that caused it
Diagnostic Coverage	It is a fraction of Dangerous Failures detected by automatic online Diagnostic tests. The fraction of Dangerous Failures is computed by using the Dangerous Failure rates associated with the detected Dangerous Failures divided by the total rate of Dangerous Failures.
Error	It is the discrepancy between a computed, observed or measured value or condition and the true, specified or theoretically correct value or condition
Failure	It is the termination of the ability of a functional unit to provide a required function or operation of a functional unit in any way other than as required
Failure Rate (λ(t))	It is a reliability parameter (λ(t)) of an entity (single components or systems), such that λ(t).dt is the probability of Failure of this entity within [t, t+dt] provided that it has not failed during [0, t]
Fault	It is an abnormal condition that may cause a reduction in, or loss of, the capability of a functional unit to perform a required function

Fault Avoidance	It defines the use of techniques and procedures that aim to avoid the introduction of Faults during any phase of the Safety life cycle of the Safety-related system
Fault Tolerance	It is the ability of a functional unit to continue to perform a required function in the presence of Faults or errors
Probability of Dangerous Failure on Demand (PFD)	It is the Safety unavailability of a Safety-related system to perform the specified Safety Function when a demand occurs
Random Hardware Failure	It is the Failure, occurring at a random time, which results from one or more of the possible degradation mechanisms in the hardware
Safe Failure	It is the Failure of an element/subsystem/system that plays a part in implementing the Safety Function that: a) Results in the spurious operation of the Safety Function to put (part of) the EUC into a Safe State or maintain a Safe State; or b) Increases the probability of the spurious operation of the Safety Function to put (part of) the EUC into a Safe State or maintain a Safe State.
Safe Failure Fraction (SFF)	It is the property of a Safety-related element that is defined by the ratio of the average Failure Rates of Safe plus Dangerous detected Failures and Safe plus Dangerous Failures. This ratio is represented by the following equation: $SFF = (\Sigma\lambda_{S\ avg} + \Sigma\lambda_{Dd\ avg})/(\Sigma\lambda_{S\ avg} + \Sigma\lambda_{Dd\ avg} + \Sigma\lambda_{Du\ avg}))$ Where: S = Safe, Dd = Dangerous Detected, Du = Dangerous Undetected. When the Failure Rates are based on constant Failure Rates, the equation can be simplified to: $SFF = (\Sigma\lambda_S + \Sigma\lambda_{Dd})/(\Sigma\lambda_S + \Sigma\lambda_{Dd} + \Sigma\lambda_{Du})$
Systematic Failure	It is the Failure, related in a deterministic way to a certain cause, which can only be eliminated by a modification of the design or of the manufacturing process, operational procedures, documentation or other relevant factors
Target Failure Measure	It is the target probability of Dangerous mode Failures to be achieved in respect of the Safety integrity requirements specified in terms of either: • The average probability of a Dangerous Failure of the Safety Function on demand (for a Low Demand mode of operation) • the average frequency of a Dangerous Failure [h^{-1}] (for a High Demand or Continuous-mode operation)

Table 1-9. Keywords – Degree of Obligation

Keyword	Degree of Obligation
Shall	Binding
Should	Recommended

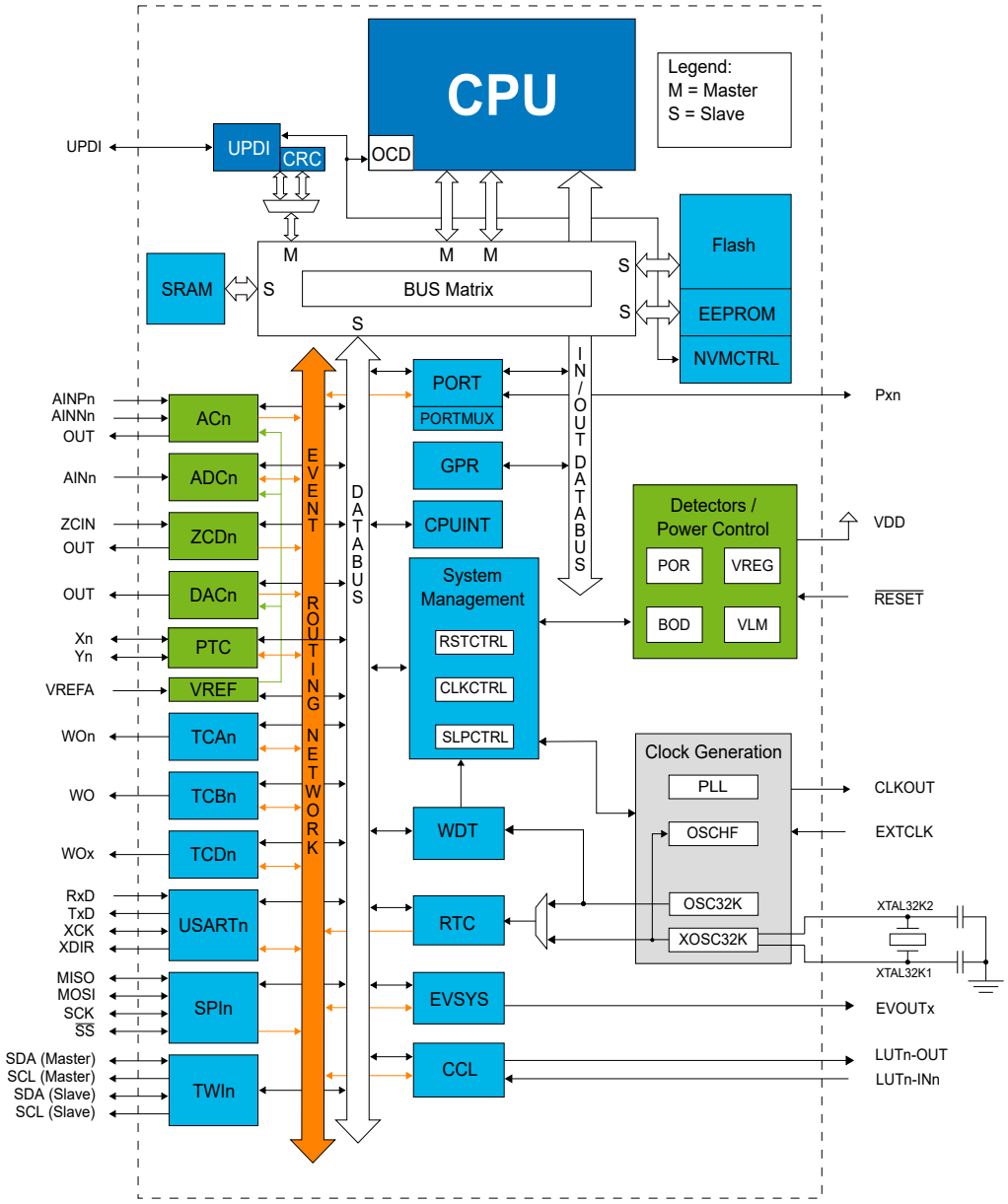
2. Product Overview

The MCU family are 5.5V microcontrollers with up to 128 KB of Flash, 512B of EEPROM and 16 KB of SRAM. The figure below shows a general block diagram of the core and peripheral modules.

Some of the features listed in this manual may not apply to a specific device in the device family. The System Integrator is expected to review the data sheet of the device of interest to verify the availability and the number of the peripherals of interest.

Refer to the device data sheet for additional product details.

Figure 2-1. General Block Diagram of CPU Core and Peripherals



2.1 Device Intended Use

The Microchip devices are general purpose microcontrollers for automotive and industrial applications, as well as household appliances, though they may be used for other types of applications as well. The devices supporting

the Peripheral Touch Controller (PTC) is intended to be used for interfacing capacitive touch sensors and the implementation of capacitive touch user interfaces, such as buttons, sliders, wheels and 2D surfaces.

Targeted automotive applications utilizing the device's PTC are:

- In-cabin capacitive sensors for proximity or touch detection
- Exterior capacitive sensors for proximity or touch detection

The Microchip devices are considered a Safety Element out of Context (SEooC). The assumptions of use are based on the use as a general purpose microcontroller and specifically as a capacitive touch sensor controller.

The Top-level Safety Requirement (TLSR) for these devices is defined as: Correct data acquisition, processing and resulting actuation, as well as communication with a secondary or higher-ranking system, if one exists. Any fault in a safety-related block identified by the System Integrator is considered a TLSR violation. Refer to the Assumptions of Use sections.

2.2 Supported Devices

MCU4	MCU3	MCU2	MCU1

2.3 ISO 26262 – Mission Profile

Important: This section is relevant for ISO 26262 only. The content can still be of interest for IEC 61508, but in this case, it describes assumed application conditions.

The key aspects of the mission profile are summarized in the following table.

Target	Value	Comments
Device Lifetime	15 years	15 years correspond to 10000 hours of active operation of the MCU. It might be necessary to adjust the device lifetime per the specific operating profile of the application.
Safety Function Target	ASIL B	This is the target of the safety functions for this device
Total Active Operation	10000 hours	
Maximum Time of Operation without a Start-up Reset	10 hours	
Relevant Ambient Temperature in Use Environment	Variable	Use the FMEDA with the relevant application temperature for FIT Rate calculation
FTTI Budget	Application-dependent	The required FTTI period depends on the application. The System Integrator determines the required FTTI period and implements the Safety Mechanisms meeting the FTTI requirement.

Note: Diagnostic methods described within and coverage calculations obtained from the FMEDA can be used as a guide to determine additional compliance levels.

2.4 Operating Conditions

The table below lists the various operating conditions.

The System Integrator shall comply with the operating conditions for the device of interest defined in the data sheet.

Table 2-1. Operating Conditions

Temperature Range	Supply Range	Clock Speed	Qualifications
-40°C to +85°C	1.8V-5.5V	DC to 24 MHz	AEC-Q100 revision H, Grade 3
-40°C to +125°C	1.8V-5.5V	DC to 20 MHz	AEC-Q100 revision H, Grade 1

3. Safety Measures

Safety measures are activities or technical solutions implemented to:

- Avoid or Control Systematic Failures
- Detect Random Hardware Failures
- Control Random Hardware Failures
- Mitigate the Harmful Effects of Failures

Safety measures are described in Section "Diagnostic Mechanisms" and summarized in the tables below.

Table 3-1. Processing Units Diagnostic Mechanisms

ISO 26262 – Table D.4 Processing Units			IEC 61508-7, Edition 2.0 2010-04 – Table A-4 Processing Units			Safety Measures Available from Microchip
Safety Mechanism/Measure	Reference(1)	Coverage(2)	Diagnostic Technique/Measure	Reference(3)	Coverage(4)	
Self-test by software: limited number of patterns (one channel)	D.2.3.1	Medium (90%)	Self-test by software: limited number of patterns (one channel)	A.3.1	Low (60%)	No Safety Measure is described in this manual
N/A	—	—	Self-test by software: walking bit (one channel)	A.3.2	Medium (90%)	No Safety Measure described in this manual
Self-test supported by hardware (one channel)	D.2.3.2	Medium (90%)	Self-test supported by hardware (one channel)	A.3.3	Medium (90%)	No Safety Measure is described in this manual
Self-test by software cross exchange between two independent units	D.2.3.3	Medium (90%)	N/A			No Safety Measure described in this manual
Software diversified redundancy (one hardware channel)	D.2.3.4	High (99%)	N/A			No Safety Measure is described in this manual
Reciprocal comparison by software	D.2.3.5	High (99%)	Reciprocal comparison by software	A.3.5	High (99%)	No Safety Measure described in this manual
HW redundancy (e.g., dual core lockstep, asymmetric redundancy, coded processing)	D.2.3.6	High (99%)	Coded processing (one channel)	A.3.4	High (99%)	No Safety Measure described in this manual
Configuration register test	D.2.3.7	High (99%)	N/A			IO_REGISTER_RESET_STATE_CHECK IO_REGISTER_WRITE_READ_TEST
Stack over/underflow detection	D.2.3.8	Low (60%)	N/A			STACK_GUARD_PATTERN_TEST
Integrated hardware consistency monitoring	D.2.3.9	High (99%)	N/A			No Safety Measure is described in this manual
N/A	—	—	Comparator	A.1.3	High (99%)	No Safety Measure described in this manual
N/A	—	—	Majority voter	A.1.4	High (99%)	No Safety Measure described in this manual

Notes:

1. Refer to ISO 26262-5:2018.
2. Typical diagnostic coverage considered achievable.
3. Refer to IEC 61508-7.
4. Maximum diagnostic coverage considered achievable.

Table 3-2. Analog and Digital I/O Diagnostic Mechanisms

ISO 26262-11:2018 – Table D.5 Analog and Digital I/Os			IEC 61508-7, Edition 2.0 2010-4 – Table A.7 I/O Units and Interface (External Communication)			Safety Measures Available from Microchip
Safety Mechanism/Measure	Reference(1)	Coverage(2)	Diagnostic Technique/Measure	Reference(3)	Coverage(4)	
Failure detection by online monitoring (digital I/Os)	D.2.1.1	Low (60%)	Failure detection by on-line monitoring	A.1.1	Low (60%) (Low Demand mode)	No Safety Measure described in this manual
				A.1.1	Medium (90%) (High Demand or Continuous mode)	No Safety Measure described in this manual
Test pattern	D.2.4.1	High (99%)	Test pattern	A.6.1	High (99%)	No Safety Measure is described in this manual
Code protection for digital I/Os	D.2.4.2	Medium (90%)	Code protection	A.6.2	High (99%)	IO_PORTS_CHANGE_NOTIFICATION_TEST(5)
Multichannel parallel output	D.2.4.3	High (99%)	Multichannel parallel output	A.6.3	High (99%)	No Safety Measure described in this manual

.....continued

ISO 26262-11:2018 – Table D.5 Analog and Digital I/Os			IEC 61508-7, Edition 2.0 2010-4 – Table A.7 I/O Units and Interface (External Communication)			Safety Measures Available from Microchip
Safety Mechanism/Measure	Reference(1)	Coverage(2)	Diagnostic Technique/Measure	Reference(3)	Coverage(4)	
Monitored outputs	D.2.4.4	High (99%)	Monitored outputs	A.6.4	High (99%)	IO_PORTS_OUTPUT_MONITOR
Input comparison/voting (1oo2, 2oo3 or better redundancy)	D.2.4.5	High (99%)	Input comparison/voting (1oo2, 2oo3 or better redundancy)	A.6.5	High (99%)	IO_PORTS_INPUT_COMPARISON
N/A			Antivalent signal transmission	A.11.4	High (99%)	No Safety Measure described in this manual

Notes:

1. Refer to ISO 26262-5:2018.
2. Typical diagnostic coverage considered achievable.
3. Refer to IEC 61508-7.
4. Maximum diagnostic coverage considered achievable.

Table 3-3. Program Sequence Monitoring/Clock Diagnostic Mechanisms

ISO 26262-5:2018 – Table D.8 Program Sequence Monitoring/Clock			IEC 61508-7, Edition 2.0 2010-04 – Table A.10 Program Sequence (Watchdog), Table A.11 Clock			Safety Measures Available from Microchip
Safety Mechanism/Measure	Reference(1)	Coverage(2)	Diagnostic Technique/Measure	Reference(3)	Coverage(4)	
Watchdog with separate time base without time window	D.2.7.1	Low (60%)	Watchdog with separate time base without time window	A.9.1	Low (60%)	No Safety Measure described in this manual
Watchdog with separate time base and time window	D.2.7.2	Medium (90%)	Watchdog with separate time base and time window	A.9.2	Medium (90%)	No Safety Measure described in this manual
Logical monitoring of program sequence	D.2.7.3	Medium (90%)	Logical monitoring of program sequence	A.9.3	Medium (90%)	No Safety Measure is described in this manual
Combination of temporal and logical monitoring of program sequences	D.2.7.4	High (99%)	Combination of temporal and logical monitoring of program sequences	A.9.4	High (99%)	CLOCK_CONTINUOUS_EXTERNAL_MONITOR
Combination of temporal and logical monitoring of program sequences with time dependency	D.2.7.5	High (99%)	N/A			CLOCK_PERIODIC_MONITOR
N/A			Temporal monitoring with on-line check	A.9.5	Medium (90%)	No Safety Measure described in this manual

Notes:

1. Refer to ISO 26262-5:2018.
2. Typical diagnostic coverage considered achievable.
3. Refer to IEC 61508-7.
4. Maximum diagnostic coverage considered achievable.

Table 3-4. Nonvolatile Memory Diagnostic Mechanisms

ISO 26262-11:2018 – Table 32 Non-Volatile Memory			IEC 61508-7, Edition 2.0 2010-04 – Table A.5 Invariable Memory Ranges			Safety Measures Available from Microchip
Safety Mechanism/Measure	Reference(1)	Coverage(2)	Diagnostic Technique/Measure	Reference(3)	Coverage(4)	
Parity Bit	5.1.13.6	Low (60%)	Word-protection multibit redundancy	A.4.1	Medium (90%)	No Safety Measure described in this manual
Memory Monitoring using Error-Detection-Correction (EDC) Codes	5.1.13.1	High (99%)	N/A			No Safety Measure described in this manual
Modified Checksum	5.1.13.2	Low (60%)	Modified checksum	A.4.2	Low (60%)	No Safety Measure described in this manual
Memory Signature	5.1.13.3	High (99%)	Signature of one word (8-bit)	A.4.3	Medium (90%)	FLASH_MEMORY_CHECKSUM_CRC_TEST FLASH_MEMORY_CRCSCAN_TEST EEPROM_MEMORY_CHECKSUM_CRC_TEST
			Signature of a double word (16-bit)	A.4.4	High (99%)	
Block Replication	5.1.13.4	High (99%)	Block Replication	A.4.5	High (99%)	FLASH_MEMORY_BLOCK_REPLICATION EEPROM_MEMORY_BLOCK_REPLICATION

.....continued

ISO 26262-11:2018 – Table 32 Non-Volatile Memory			IEC 61508-7, Edition 2.0 2010-04 – Table A.5 Invariable Memory Ranges			Safety Measures Available from Microchip
Safety Mechanism/Measure	Reference(1)	Coverage(2)	Diagnostic Technique/Measure	Reference(3)	Coverage(4)	

Notes:

1. Refer to ISO 26262-11:2018.
2. Typical diagnostic coverage considered achievable.
3. Refer to IEC 61508-7.
4. Maximum diagnostic coverage considered achievable.

Table 3-5. Volatile Memory Diagnostic Mechanisms

ISO 26262-11:2018 – Table 33 Volatile Memory			IEC 61508-7, Edition 2.0 2010-04 – Table A.6 Variable Memory Ranges			Safety Measures Available from Microchip
Safety Mechanism/Measure	Reference(1)	Coverage(2)	Diagnostic Technique/Measure	Reference(3)	Coverage(4)	
Memory monitoring using Error Correction Codes (ECC)	5.1.13.1	High (99%)	RAM monitoring with a modified Hamming code or detection of data failures with Error Detection-Correction Codes (EDC)	A.5.6	Medium (90%)	No Safety Measure described in this manual
Block replication	5.1.13.4	High (99%)	Double RAM with hardware or software comparison and read/ write test	A.5.7	High (99%)	IO_REGISTER_BLOCK_REPLICATION SRAM_DUPLICATION STACK_CONTEXT_INTEGRITY_CHECK
RAM pattern test	5.1.13.5	Medium (90%)	N/A			No Safety Measure described in this manual
Parity bit	5.1.13.6	Low (60%)	Parity bit for RAM	A.5.5	Low (60%)	No Safety Measure described in this manual
RAM march test	5.1.13.7	High (99%)	N/A			SRAM_MARCH_TEST
Running checksum/CRC	5.1.13.8	High (99%)	N/A			No Safety Measure described in this manual
N/A			RAM test checkerboard or march	A.5.1	Low (60%)	No Safety Measure described in this manual
N/A			RAM test walk path	A.5.2	Medium (90%)	No Safety Measure described in this manual
N/A			RAM test galpat or transparent galpat	A.5.3	High (99%)	SRAM_MARCH_TEST**(5)**
N/A			RAM test abraham	A.5.4	High (99%)	SRAM_MARCH_TEST**(5)**

Notes:

1. Refer to ISO 26262-11:2018.
2. Typical diagnostic coverage considered achievable.
3. Refer to IEC 61508-7.
4. Maximum diagnostic coverage considered achievable.
5. The test implements the March C Minus algorithm, which has the same or higher coverage than galpat, transparent galpat, and abraham RAM tests.

4. Diagnostic Mechanisms

This section provides a summary of the Diagnostic Mechanisms considered in the FMEDA. Refer to the device data sheet for details on the device features/peripherals.

The information is organized in the form of tables (one for each Diagnostic Mechanism). The tables have two sections:

- The first section collects all the relevant information to allow the System Integrator to understand and use the Diagnostic Mechanism
- The second section correlates the Diagnostic Mechanism to specific content of the ISO 26262-5:2018 and ISO 26262-11:2018. It also correlates to the specific content of the IEC 61508-2:2010 and IEC 61508-7:2010.

The content of the first section of the tables is described in the table below.

Table 4-1. Diagnostic Mechanism Information

Purpose	A short explanation of what issue the Diagnostic Mechanism is addressing and how it is implemented
Description	Additional information to clarify the details of the implementation
Initialization/Setup	Actions needed to make sure the Diagnostic Mechanism, as envisioned, works correctly. Usually, this refers to register or operational mode settings. In case of additional device resources used for the Diagnostic Mechanism implementation, Initialization only refers to these resources. The Initialization of the peripheral used in the application is application-dependent and is the responsibility of the System Integrator.
Error Reporting	A mechanism used to convey the result of the diagnostic test to the application code calling the diagnostic routine
Diagnostic Type	Describes how the test result is obtained. There are two options: • **Software** = The test result is obtained with a software operation (for example, RAM locations comparison) • **Hardware** = The test result is obtained through hardware (for example, external supervisor circuit used to monitor V_{DD}) Note that there are also two additional sub-options: • Software requiring hardware support (for example, external loopback tests) • Hardware requiring software support (for example, period monitoring)
Recommended Mitigation Action	Possible action to be performed to reduce/eliminate the impact of the failure addressed by the Diagnostic Mechanism

Periodicity	Describes when the routine should be executed. In general, some of the routines are “destructive” in the sense that they will corrupt the content of the memory and/or registers; others are not. Four possible options are considered in this manual: • **On start-up:** The case when running the diagnostics will corrupt the content of the memory/registers, or the time required to run the diagnostics is so long that it is unlikely that it can be run during the normal operation of the device when executing the System Integrator’s application code. • **On demand:** This is typically not a destructive mechanism, as it allows the system to operate without interruption. The diagnostic measure is run every time the device feature is used. • **Periodic:** This is typically not a destructive mechanism, as it allows the system to operate without interruption. The System Integrator will call the diagnostic routine at specific instants in time based on the application. This option allows the System Integrator to run diagnostics that require relevant execution times at reasonable instants that do not impact the behavior and performance of the application. • **Continuous:** This is typically not a destructive mechanism, as it allows the system to operate without interruption. Some diagnostics, by their nature, can be/are run continuously without impacting the System Integrator’s application behavior or performance. Note that the table provides suggestions. The final decision of the periodicity is the responsibility of the System Integrator.
Recommendations and Limitations	Notes and information useful to correctly use the mechanism and to have a correct understanding of its limitations (in use and coverage), as needed
Relevant Software Requirements	The name of any specific software required to implement the Diagnostic Mechanism. The software requirements are listed in 6. Software Requirements on System Level.
Relevant Hardware Requirements	The name of any external hardware (for example, connections between device pins) required to implement the Diagnostic Mechanism. The hardware requirements are listed and described in 5. Hardware Requirements on System Level.
Dataflow Dependency	Highlights if the diagnostic tests operate on ad hoc data or use the data that are normally managed by the application while running. There are two options: • **Dependent:** The tests and their results depend on the data being used by the device during the application code execution. • **Independent:** The tests and their results depend on the data that are specified by the tests themselves. Usually, this is the case for diagnostics run on start-up (see Periodicity above).
Reference to ISO 26262:2018 Contents Section	
ISO 26262-x Table	Identifies the table in ISO 26262-5:2018 or ISO 26262-11:2018 that lists the Safety Mechanism/Measure implemented by the Diagnostic Mechanism; “x” is either “5” or “11”. ISO 26262-5:2018, Annex D contains Tables D.1 through D.10. ISO 26262-11:2018, Sections 5.1.12., 5.2.2 and 5.2.4 contain Tables 32 through 40.
ISO 26262-x Paragraph	Identifies the technique described in ISO 26262-5:2018 or ISO 26262-11:2018 that is implemented by the Diagnostic Mechanism; “x” is either “5” or “11”
Safety Mechanism Measure	Name of the mechanism as per ISO 26262-5:2018 and ISO 26262-11:2018

Assessed Diagnostic Coverage	Identifies a percentage of the coverage that can be achieved using the corresponding measure. It has three possible values: • Low = 60% • Medium = 90% • High = 99% These values are used in the overall FMEDA computations for determining the SIL/ASIL level.
Reference to IEC 61508, Edition 2.0 2010-04 Section	
61508-2 © IEC:2010 – Table	Identifies the 61508-2, IEC:2010 table that includes the Safety Mechanism/Measure implemented by the routine. The IEC:2010 tables are in Part 2, Annex A12.
61508-7 © IEC:2010 – Paragraph	Identifies the 61508-7 technique for embedded Diagnostic self-tests
Diagnostic Technique/ Measure	Reflects the mechanism name as per 61508-2 © IEC:2010
Assessed Diagnostic Coverage	Identifies a percentage of the coverage that can be achieved using the corresponding measure. It has three possible values: • Low = 60% • Medium = 90% • High = 99% These values are used in the overall FMEDA computations for determining the SIL/ASIL level.

4.1 Clocks

4.1.1 Diagnostics Mechanism

Table 4-2. Diagnostic Mechanism Information

Purpose	A short explanation of what issue the Diagnostic Mechanism is addressing and how it is implemented
Description	Additional information to clarify the details of the implementation
Initialization/Setup	Actions needed to make sure the Diagnostic Mechanism, as envisioned, works correctly. Usually, this refers to register or operational mode settings. In case of additional device resources used for the Diagnostic Mechanism implementation, Initialization only refers to these resources. The Initialization of the peripheral used in the application is application-dependent and is the responsibility of the System Integrator.
Error Reporting	A mechanism used to convey the result of the diagnostic test to the application code calling the diagnostic routine
Diagnostic Type	Describes how the test result is obtained. There are two options: • **Software** = The test result is obtained with a software operation (for example, RAM locations comparison) • **Hardware** = The test result is obtained through hardware (for example, external supervisor circuit used to monitor V_{DD}) Note that there are also two additional sub-options: • Software requiring hardware support (for example, external loopback tests) • Hardware requiring software support (for example, period monitoring)

Recommended Mitigation Action	Possible action to be performed to reduce/eliminate the impact of the failure addressed by the Diagnostic Mechanism
Periodicity	Describes when the routine should be executed. In general, some of the routines are “destructive” in the sense that they will corrupt the content of the memory and/or registers; others are not. Four possible options are considered in this manual: • **On start-up:** The case when running the diagnostics will corrupt the content of the memory/registers, or the time required to run the diagnostics is so long that it is unlikely that it can be run during the normal operation of the device when executing the System Integrator’s application code. • **On demand:** This is typically not a destructive mechanism, as it allows the system to operate without interruption. The diagnostic measure is run every time the device feature is used. • **Periodic:** This is typically not a destructive mechanism, as it allows the system to operate without interruption. The System Integrator will call the diagnostic routine at specific instants in time based on the application. This option allows the System Integrator to run diagnostics that require relevant execution times at reasonable instants that do not impact the behavior and performance of the application. • **Continuous:** This is typically not a destructive mechanism, as it allows the system to operate without interruption. Some diagnostics, by their nature, can be/are run continuously without impacting the System Integrator’s application behavior or performance. Note that the table provides suggestions. The final decision of the periodicity is the responsibility of the System Integrator.
Recommendations and Limitations	Notes and information useful to correctly use the mechanism and to have a correct understanding of its limitations (in use and coverage), as needed
Relevant Software Requirements	The name of any specific software required to implement the Diagnostic Mechanism. The software requirements are listed in 6. Software Requirements on System Level.
Relevant Hardware Requirements	The name of any external hardware (for example, connections between device pins) required to implement the Diagnostic Mechanism. The hardware requirements are listed and described in 5. Hardware Requirements on System Level.
Dataflow Dependency	Highlights if the diagnostic tests operate on ad hoc data or use the data that are normally managed by the application while running. There are two options: • **Dependent:** The tests and their results depend on the data being used by the device during the application code execution. • **Independent:** The tests and their results depend on the data that are specified by the tests themselves. Usually, this is the case for diagnostics run on start-up (see Periodicity above).
Reference to ISO 26262:2018 Contents Section	
ISO 26262-x Table	Identifies the table in ISO 26262-5:2018 or ISO 26262-11:2018 that lists the Safety Mechanism/Measure implemented by the Diagnostic Mechanism; “x” is either “5” or “11”. ISO 26262-5:2018, Annex D contains Tables D.1 through D.10. ISO 26262-11:2018, Sections 5.1.12., 5.2.2 and 5.2.4 contain Tables 32 through 40.
ISO 26262-x Paragraph	Identifies the technique described in ISO 26262-5:2018 or ISO 26262-11:2018 that is implemented by the Diagnostic Mechanism; “x” is either “5” or “11”
Safety Mechanism Measure	Name of the mechanism as per ISO 26262-5:2018 and ISO 26262-11:2018

Assessed Diagnostic Coverage	Identifies a percentage of the coverage that can be achieved using the corresponding measure. It has three possible values: • Low = 60% • Medium = 90% • High = 99% These values are used in the overall FMEDA computations for determining the SIL/ASIL level.
Reference to IEC 61508, Edition 2.0 2010-04 Section	
61508-2 © IEC:2010 – Table	Identifies the 61508-2, IEC:2010 table that includes the Safety Mechanism/Measure implemented by the routine. The IEC:2010 tables are in Part 2, Annex A12.
61508-7 © IEC:2010 – Paragraph	Identifies the 61508-7 technique for embedded Diagnostic self-tests
Diagnostic Technique/ Measure	Reflects the mechanism name as per 61508-2 © IEC:2010
Assessed Diagnostic Coverage	Identifies a percentage of the coverage that can be achieved using the corresponding measure. It has three possible values: • Low = 60% • Medium = 90% • High = 99% These values are used in the overall FMEDA computations for determining the SIL/ASIL level.

4.2 EEPROM

4.2.1 Diagnostics Mechanism

Table 4-3. Diagnostic Mechanism Information

Purpose	A short explanation of what issue the Diagnostic Mechanism is addressing and how it is implemented
Description	Additional information to clarify the details of the implementation
Initialization/Setup	Actions needed to make sure the Diagnostic Mechanism, as envisioned, works correctly. Usually, this refers to register or operational mode settings. In case of additional device resources used for the Diagnostic Mechanism implementation, Initialization only refers to these resources. The Initialization of the peripheral used in the application is application-dependent and is the responsibility of the System Integrator.
Error Reporting	A mechanism used to convey the result of the diagnostic test to the application code calling the diagnostic routine
Diagnostic Type	Describes how the test result is obtained. There are two options: • **Software** = The test result is obtained with a software operation (for example, RAM locations comparison) • **Hardware** = The test result is obtained through hardware (for example, external supervisor circuit used to monitor V_{DD}) Note that there are also two additional sub-options: • Software requiring hardware support (for example, external loopback tests) • Hardware requiring software support (for example, period monitoring)

Recommended Mitigation Action	Possible action to be performed to reduce/eliminate the impact of the failure addressed by the Diagnostic Mechanism
Periodicity	Describes when the routine should be executed. In general, some of the routines are “destructive” in the sense that they will corrupt the content of the memory and/or registers; others are not. Four possible options are considered in this manual: • **On start-up:** The case when running the diagnostics will corrupt the content of the memory/registers, or the time required to run the diagnostics is so long that it is unlikely that it can be run during the normal operation of the device when executing the System Integrator’s application code. • **On demand:** This is typically not a destructive mechanism, as it allows the system to operate without interruption. The diagnostic measure is run every time the device feature is used. • **Periodic:** This is typically not a destructive mechanism, as it allows the system to operate without interruption. The System Integrator will call the diagnostic routine at specific instants in time based on the application. This option allows the System Integrator to run diagnostics that require relevant execution times at reasonable instants that do not impact the behavior and performance of the application. • **Continuous:** This is typically not a destructive mechanism, as it allows the system to operate without interruption. Some diagnostics, by their nature, can be/are run continuously without impacting the System Integrator’s application behavior or performance. Note that the table provides suggestions. The final decision of the periodicity is the responsibility of the System Integrator.
Recommendations and Limitations	Notes and information useful to correctly use the mechanism and to have a correct understanding of its limitations (in use and coverage), as needed
Relevant Software Requirements	The name of any specific software required to implement the Diagnostic Mechanism. The software requirements are listed in 6. Software Requirements on System Level.
Relevant Hardware Requirements	The name of any external hardware (for example, connections between device pins) required to implement the Diagnostic Mechanism. The hardware requirements are listed and described in 5. Hardware Requirements on System Level.
Dataflow Dependency	Highlights if the diagnostic tests operate on ad hoc data or use the data that are normally managed by the application while running. There are two options: • **Dependent:** The tests and their results depend on the data being used by the device during the application code execution. • **Independent:** The tests and their results depend on the data that are specified by the tests themselves. Usually, this is the case for diagnostics run on start-up (see Periodicity above).
Reference to ISO 26262:2018 Contents Section	
ISO 26262-x Table	Identifies the table in ISO 26262-5:2018 or ISO 26262-11:2018 that lists the Safety Mechanism/Measure implemented by the Diagnostic Mechanism; “x” is either “5” or “11”. ISO 26262-5:2018, Annex D contains Tables D.1 through D.10. ISO 26262-11:2018, Sections 5.1.12., 5.2.2 and 5.2.4 contain Tables 32 through 40.
ISO 26262-x Paragraph	Identifies the technique described in ISO 26262-5:2018 or ISO 26262-11:2018 that is implemented by the Diagnostic Mechanism; “x” is either “5” or “11”
Safety Mechanism Measure	Name of the mechanism as per ISO 26262-5:2018 and ISO 26262-11:2018

Assessed Diagnostic Coverage	Identifies a percentage of the coverage that can be achieved using the corresponding measure. It has three possible values: • Low = 60% • Medium = 90% • High = 99% These values are used in the overall FMEDA computations for determining the SIL/ASIL level.
Reference to IEC 61508, Edition 2.0 2010-04 Section	
61508-2 © IEC:2010 – Table	Identifies the 61508-2, IEC:2010 table that includes the Safety Mechanism/Measure implemented by the routine. The IEC:2010 tables are in Part 2, Annex A12.
61508-7 © IEC:2010 – Paragraph	Identifies the 61508-7 technique for embedded Diagnostic self-tests
Diagnostic Technique/ Measure	Reflects the mechanism name as per 61508-2 © IEC:2010
Assessed Diagnostic Coverage	Identifies a percentage of the coverage that can be achieved using the corresponding measure. It has three possible values: • Low = 60% • Medium = 90% • High = 99% These values are used in the overall FMEDA computations for determining the SIL/ASIL level.

4.3 Flash Memory

4.3.1 Diagnostics Mechanism

Table 4-4. Diagnostic Mechanism Information

Purpose	A short explanation of what issue the Diagnostic Mechanism is addressing and how it is implemented
Description	Additional information to clarify the details of the implementation
Initialization/Setup	Actions needed to make sure the Diagnostic Mechanism, as envisioned, works correctly. Usually, this refers to register or operational mode settings. In case of additional device resources used for the Diagnostic Mechanism implementation, Initialization only refers to these resources. The Initialization of the peripheral used in the application is application-dependent and is the responsibility of the System Integrator.
Error Reporting	A mechanism used to convey the result of the diagnostic test to the application code calling the diagnostic routine
Diagnostic Type	Describes how the test result is obtained. There are two options: • **Software** = The test result is obtained with a software operation (for example, RAM locations comparison) • **Hardware** = The test result is obtained through hardware (for example, external supervisor circuit used to monitor V_{DD}) Note that there are also two additional sub-options: • Software requiring hardware support (for example, external loopback tests) • Hardware requiring software support (for example, period monitoring)

Recommended Mitigation Action	Possible action to be performed to reduce/eliminate the impact of the failure addressed by the Diagnostic Mechanism
Periodicity	Describes when the routine should be executed. In general, some of the routines are “destructive” in the sense that they will corrupt the content of the memory and/or registers; others are not. Four possible options are considered in this manual: • **On start-up:** The case when running the diagnostics will corrupt the content of the memory/registers, or the time required to run the diagnostics is so long that it is unlikely that it can be run during the normal operation of the device when executing the System Integrator’s application code. • **On demand:** This is typically not a destructive mechanism, as it allows the system to operate without interruption. The diagnostic measure is run every time the device feature is used. • **Periodic:** This is typically not a destructive mechanism, as it allows the system to operate without interruption. The System Integrator will call the diagnostic routine at specific instants in time based on the application. This option allows the System Integrator to run diagnostics that require relevant execution times at reasonable instants that do not impact the behavior and performance of the application. • **Continuous:** This is typically not a destructive mechanism, as it allows the system to operate without interruption. Some diagnostics, by their nature, can be/are run continuously without impacting the System Integrator’s application behavior or performance. Note that the table provides suggestions. The final decision of the periodicity is the responsibility of the System Integrator.
Recommendations and Limitations	Notes and information useful to correctly use the mechanism and to have a correct understanding of its limitations (in use and coverage), as needed
Relevant Software Requirements	The name of any specific software required to implement the Diagnostic Mechanism. The software requirements are listed in 6. Software Requirements on System Level.
Relevant Hardware Requirements	The name of any external hardware (for example, connections between device pins) required to implement the Diagnostic Mechanism. The hardware requirements are listed and described in 5. Hardware Requirements on System Level.
Dataflow Dependency	Highlights if the diagnostic tests operate on ad hoc data or use the data that are normally managed by the application while running. There are two options: • **Dependent:** The tests and their results depend on the data being used by the device during the application code execution. • **Independent:** The tests and their results depend on the data that are specified by the tests themselves. Usually, this is the case for diagnostics run on start-up (see Periodicity above).
Reference to ISO 26262:2018 Contents Section	
ISO 26262-x Table	Identifies the table in ISO 26262-5:2018 or ISO 26262-11:2018 that lists the Safety Mechanism/Measure implemented by the Diagnostic Mechanism; “x” is either “5” or “11”. ISO 26262-5:2018, Annex D contains Tables D.1 through D.10. ISO 26262-11:2018, Sections 5.1.12., 5.2.2 and 5.2.4 contain Tables 32 through 40.
ISO 26262-x Paragraph	Identifies the technique described in ISO 26262-5:2018 or ISO 26262-11:2018 that is implemented by the Diagnostic Mechanism; “x” is either “5” or “11”
Safety Mechanism Measure	Name of the mechanism as per ISO 26262-5:2018 and ISO 26262-11:2018

Assessed Diagnostic Coverage	Identifies a percentage of the coverage that can be achieved using the corresponding measure. It has three possible values: • Low = 60% • Medium = 90% • High = 99% These values are used in the overall FMEDA computations for determining the SIL/ASIL level.
Reference to IEC 61508, Edition 2.0 2010-04 Section	
61508-2 © IEC:2010 – Table	Identifies the 61508-2, IEC:2010 table that includes the Safety Mechanism/Measure implemented by the routine. The IEC:2010 tables are in Part 2, Annex A12.
61508-7 © IEC:2010 – Paragraph	Identifies the 61508-7 technique for embedded Diagnostic self-tests
Diagnostic Technique/ Measure	Reflects the mechanism name as per 61508-2 © IEC:2010
Assessed Diagnostic Coverage	Identifies a percentage of the coverage that can be achieved using the corresponding measure. It has three possible values: • Low = 60% • Medium = 90% • High = 99% These values are used in the overall FMEDA computations for determining the SIL/ASIL level.

4.4 General Purpose

4.4.1 Diagnostics Mechanism

Table 4-5. Diagnostic Mechanism Information

Purpose	A short explanation of what issue the Diagnostic Mechanism is addressing and how it is implemented
Description	Additional information to clarify the details of the implementation
Initialization/Setup	Actions needed to make sure the Diagnostic Mechanism, as envisioned, works correctly. Usually, this refers to register or operational mode settings. In case of additional device resources used for the Diagnostic Mechanism implementation, Initialization only refers to these resources. The Initialization of the peripheral used in the application is application-dependent and is the responsibility of the System Integrator.
Error Reporting	A mechanism used to convey the result of the diagnostic test to the application code calling the diagnostic routine
Diagnostic Type	Describes how the test result is obtained. There are two options: • **Software** = The test result is obtained with a software operation (for example, RAM locations comparison) • **Hardware** = The test result is obtained through hardware (for example, external supervisor circuit used to monitor V_{DD}) Note that there are also two additional sub-options: • Software requiring hardware support (for example, external loopback tests) • Hardware requiring software support (for example, period monitoring)

Recommended Mitigation Action	Possible action to be performed to reduce/eliminate the impact of the failure addressed by the Diagnostic Mechanism
Periodicity	Describes when the routine should be executed. In general, some of the routines are “destructive” in the sense that they will corrupt the content of the memory and/or registers; others are not. Four possible options are considered in this manual: • **On start-up:** The case when running the diagnostics will corrupt the content of the memory/registers, or the time required to run the diagnostics is so long that it is unlikely that it can be run during the normal operation of the device when executing the System Integrator’s application code. • **On demand:** This is typically not a destructive mechanism, as it allows the system to operate without interruption. The diagnostic measure is run every time the device feature is used. • **Periodic:** This is typically not a destructive mechanism, as it allows the system to operate without interruption. The System Integrator will call the diagnostic routine at specific instants in time based on the application. This option allows the System Integrator to run diagnostics that require relevant execution times at reasonable instants that do not impact the behavior and performance of the application. • **Continuous:** This is typically not a destructive mechanism, as it allows the system to operate without interruption. Some diagnostics, by their nature, can be/are run continuously without impacting the System Integrator’s application behavior or performance. Note that the table provides suggestions. The final decision of the periodicity is the responsibility of the System Integrator.
Recommendations and Limitations	Notes and information useful to correctly use the mechanism and to have a correct understanding of its limitations (in use and coverage), as needed
Relevant Software Requirements	The name of any specific software required to implement the Diagnostic Mechanism. The software requirements are listed in 6. Software Requirements on System Level.
Relevant Hardware Requirements	The name of any external hardware (for example, connections between device pins) required to implement the Diagnostic Mechanism. The hardware requirements are listed and described in 5. Hardware Requirements on System Level.
Dataflow Dependency	Highlights if the diagnostic tests operate on ad hoc data or use the data that are normally managed by the application while running. There are two options: • **Dependent:** The tests and their results depend on the data being used by the device during the application code execution. • **Independent:** The tests and their results depend on the data that are specified by the tests themselves. Usually, this is the case for diagnostics run on start-up (see Periodicity above).
Reference to ISO 26262:2018 Contents Section	
ISO 26262-x Table	Identifies the table in ISO 26262-5:2018 or ISO 26262-11:2018 that lists the Safety Mechanism/Measure implemented by the Diagnostic Mechanism; “x” is either “5” or “11”. ISO 26262-5:2018, Annex D contains Tables D.1 through D.10. ISO 26262-11:2018, Sections 5.1.12., 5.2.2 and 5.2.4 contain Tables 32 through 40.
ISO 26262-x Paragraph	Identifies the technique described in ISO 26262-5:2018 or ISO 26262-11:2018 that is implemented by the Diagnostic Mechanism; “x” is either “5” or “11”
Safety Mechanism Measure	Name of the mechanism as per ISO 26262-5:2018 and ISO 26262-11:2018

Assessed Diagnostic Coverage	Identifies a percentage of the coverage that can be achieved using the corresponding measure. It has three possible values: • Low = 60% • Medium = 90% • High = 99% These values are used in the overall FMEDA computations for determining the SIL/ASIL level.
Reference to IEC 61508, Edition 2.0 2010-04 Section	
61508-2 © IEC:2010 – Table	Identifies the 61508-2, IEC:2010 table that includes the Safety Mechanism/Measure implemented by the routine. The IEC:2010 tables are in Part 2, Annex A12.
61508-7 © IEC:2010 – Paragraph	Identifies the 61508-7 technique for embedded Diagnostic self-tests
Diagnostic Technique/ Measure	Reflects the mechanism name as per 61508-2 © IEC:2010
Assessed Diagnostic Coverage	Identifies a percentage of the coverage that can be achieved using the corresponding measure. It has three possible values: • Low = 60% • Medium = 90% • High = 99% These values are used in the overall FMEDA computations for determining the SIL/ASIL level.

4.5 PORT - I/O Pin Configuration

4.5.1 Diagnostics Mechanism

Table 4-6. Diagnostic Mechanism Information

Purpose	A short explanation of what issue the Diagnostic Mechanism is addressing and how it is implemented
Description	Additional information to clarify the details of the implementation
Initialization/Setup	Actions needed to make sure the Diagnostic Mechanism, as envisioned, works correctly. Usually, this refers to register or operational mode settings. In case of additional device resources used for the Diagnostic Mechanism implementation, Initialization only refers to these resources. The Initialization of the peripheral used in the application is application-dependent and is the responsibility of the System Integrator.
Error Reporting	A mechanism used to convey the result of the diagnostic test to the application code calling the diagnostic routine
Diagnostic Type	Describes how the test result is obtained. There are two options: • **Software** = The test result is obtained with a software operation (for example, RAM locations comparison) • **Hardware** = The test result is obtained through hardware (for example, external supervisor circuit used to monitor V_{DD}) Note that there are also two additional sub-options: • Software requiring hardware support (for example, external loopback tests) • Hardware requiring software support (for example, period monitoring)

Recommended Mitigation Action	Possible action to be performed to reduce/eliminate the impact of the failure addressed by the Diagnostic Mechanism
Periodicity	Describes when the routine should be executed. In general, some of the routines are “destructive” in the sense that they will corrupt the content of the memory and/or registers; others are not. Four possible options are considered in this manual: • **On start-up:** The case when running the diagnostics will corrupt the content of the memory/registers, or the time required to run the diagnostics is so long that it is unlikely that it can be run during the normal operation of the device when executing the System Integrator’s application code. • **On demand:** This is typically not a destructive mechanism, as it allows the system to operate without interruption. The diagnostic measure is run every time the device feature is used. • **Periodic:** This is typically not a destructive mechanism, as it allows the system to operate without interruption. The System Integrator will call the diagnostic routine at specific instants in time based on the application. This option allows the System Integrator to run diagnostics that require relevant execution times at reasonable instants that do not impact the behavior and performance of the application. • **Continuous:** This is typically not a destructive mechanism, as it allows the system to operate without interruption. Some diagnostics, by their nature, can be/are run continuously without impacting the System Integrator’s application behavior or performance. Note that the table provides suggestions. The final decision of the periodicity is the responsibility of the System Integrator.
Recommendations and Limitations	Notes and information useful to correctly use the mechanism and to have a correct understanding of its limitations (in use and coverage), as needed
Relevant Software Requirements	The name of any specific software required to implement the Diagnostic Mechanism. The software requirements are listed in 6. Software Requirements on System Level.
Relevant Hardware Requirements	The name of any external hardware (for example, connections between device pins) required to implement the Diagnostic Mechanism. The hardware requirements are listed and described in 5. Hardware Requirements on System Level.
Dataflow Dependency	Highlights if the diagnostic tests operate on ad hoc data or use the data that are normally managed by the application while running. There are two options: • **Dependent:** The tests and their results depend on the data being used by the device during the application code execution. • **Independent:** The tests and their results depend on the data that are specified by the tests themselves. Usually, this is the case for diagnostics run on start-up (see Periodicity above).
Reference to ISO 26262:2018 Contents Section	
ISO 26262-x Table	Identifies the table in ISO 26262-5:2018 or ISO 26262-11:2018 that lists the Safety Mechanism/Measure implemented by the Diagnostic Mechanism; “x” is either “5” or “11”. ISO 26262-5:2018, Annex D contains Tables D.1 through D.10. ISO 26262-11:2018, Sections 5.1.12., 5.2.2 and 5.2.4 contain Tables 32 through 40.
ISO 26262-x Paragraph	Identifies the technique described in ISO 26262-5:2018 or ISO 26262-11:2018 that is implemented by the Diagnostic Mechanism; “x” is either “5” or “11”
Safety Mechanism Measure	Name of the mechanism as per ISO 26262-5:2018 and ISO 26262-11:2018

Assessed Diagnostic Coverage	Identifies a percentage of the coverage that can be achieved using the corresponding measure. It has three possible values: • Low = 60% • Medium = 90% • High = 99% These values are used in the overall FMEDA computations for determining the SIL/ASIL level.
Reference to IEC 61508, Edition 2.0 2010-04 Section	
61508-2 © IEC:2010 – Table	Identifies the 61508-2, IEC:2010 table that includes the Safety Mechanism/Measure implemented by the routine. The IEC:2010 tables are in Part 2, Annex A12.
61508-7 © IEC:2010 – Paragraph	Identifies the 61508-7 technique for embedded Diagnostic self-tests
Diagnostic Technique/ Measure	Reflects the mechanism name as per 61508-2 © IEC:2010
Assessed Diagnostic Coverage	Identifies a percentage of the coverage that can be achieved using the corresponding measure. It has three possible values: • Low = 60% • Medium = 90% • High = 99% These values are used in the overall FMEDA computations for determining the SIL/ASIL level.

4.6 SRAM

4.6.1 Diagnostics Mechanism

Table 4-7. Diagnostic Mechanism Information

Purpose	A short explanation of what issue the Diagnostic Mechanism is addressing and how it is implemented
Description	Additional information to clarify the details of the implementation
Initialization/Setup	Actions needed to make sure the Diagnostic Mechanism, as envisioned, works correctly. Usually, this refers to register or operational mode settings. In case of additional device resources used for the Diagnostic Mechanism implementation, Initialization only refers to these resources. The Initialization of the peripheral used in the application is application-dependent and is the responsibility of the System Integrator.
Error Reporting	A mechanism used to convey the result of the diagnostic test to the application code calling the diagnostic routine
Diagnostic Type	Describes how the test result is obtained. There are two options: • **Software** = The test result is obtained with a software operation (for example, RAM locations comparison) • **Hardware** = The test result is obtained through hardware (for example, external supervisor circuit used to monitor V_{DD}) Note that there are also two additional sub-options: • Software requiring hardware support (for example, external loopback tests) • Hardware requiring software support (for example, period monitoring)

Recommended Mitigation Action	Possible action to be performed to reduce/eliminate the impact of the failure addressed by the Diagnostic Mechanism
Periodicity	Describes when the routine should be executed. In general, some of the routines are “destructive” in the sense that they will corrupt the content of the memory and/or registers; others are not. Four possible options are considered in this manual: • **On start-up:** The case when running the diagnostics will corrupt the content of the memory/registers, or the time required to run the diagnostics is so long that it is unlikely that it can be run during the normal operation of the device when executing the System Integrator’s application code. • **On demand:** This is typically not a destructive mechanism, as it allows the system to operate without interruption. The diagnostic measure is run every time the device feature is used. • **Periodic:** This is typically not a destructive mechanism, as it allows the system to operate without interruption. The System Integrator will call the diagnostic routine at specific instants in time based on the application. This option allows the System Integrator to run diagnostics that require relevant execution times at reasonable instants that do not impact the behavior and performance of the application. • **Continuous:** This is typically not a destructive mechanism, as it allows the system to operate without interruption. Some diagnostics, by their nature, can be/are run continuously without impacting the System Integrator’s application behavior or performance. Note that the table provides suggestions. The final decision of the periodicity is the responsibility of the System Integrator.
Recommendations and Limitations	Notes and information useful to correctly use the mechanism and to have a correct understanding of its limitations (in use and coverage), as needed
Relevant Software Requirements	The name of any specific software required to implement the Diagnostic Mechanism. The software requirements are listed in 6. Software Requirements on System Level.
Relevant Hardware Requirements	The name of any external hardware (for example, connections between device pins) required to implement the Diagnostic Mechanism. The hardware requirements are listed and described in 5. Hardware Requirements on System Level.
Dataflow Dependency	Highlights if the diagnostic tests operate on ad hoc data or use the data that are normally managed by the application while running. There are two options: • **Dependent:** The tests and their results depend on the data being used by the device during the application code execution. • **Independent:** The tests and their results depend on the data that are specified by the tests themselves. Usually, this is the case for diagnostics run on start-up (see Periodicity above).
Reference to ISO 26262:2018 Contents Section	
ISO 26262-x Table	Identifies the table in ISO 26262-5:2018 or ISO 26262-11:2018 that lists the Safety Mechanism/Measure implemented by the Diagnostic Mechanism; “x” is either “5” or “11”. ISO 26262-5:2018, Annex D contains Tables D.1 through D.10. ISO 26262-11:2018, Sections 5.1.12., 5.2.2 and 5.2.4 contain Tables 32 through 40.
ISO 26262-x Paragraph	Identifies the technique described in ISO 26262-5:2018 or ISO 26262-11:2018 that is implemented by the Diagnostic Mechanism; “x” is either “5” or “11”
Safety Mechanism Measure	Name of the mechanism as per ISO 26262-5:2018 and ISO 26262-11:2018

Assessed Diagnostic Coverage	Identifies a percentage of the coverage that can be achieved using the corresponding measure. It has three possible values: • Low = 60% • Medium = 90% • High = 99% These values are used in the overall FMEDA computations for determining the SIL/ASIL level.
Reference to IEC 61508, Edition 2.0 2010-04 Section	
61508-2 © IEC:2010 – Table	Identifies the 61508-2, IEC:2010 table that includes the Safety Mechanism/Measure implemented by the routine. The IEC:2010 tables are in Part 2, Annex A12.
61508-7 © IEC:2010 – Paragraph	Identifies the 61508-7 technique for embedded Diagnostic self-tests
Diagnostic Technique/ Measure	Reflects the mechanism name as per 61508-2 © IEC:2010
Assessed Diagnostic Coverage	Identifies a percentage of the coverage that can be achieved using the corresponding measure. It has three possible values: • Low = 60% • Medium = 90% • High = 99% These values are used in the overall FMEDA computations for determining the SIL/ASIL level.

5. Hardware Requirements on System Level

The concept specification, the hazard and risk analysis, the overall safety requirement specification, and the consequent allocation has determined hardware requirements for the compliant item (ASR, Assumed Safety Requirements) listed below. The following points should be kept in mind while implementing these hardware requirements on a specific System Integrator's application:

- The list of hardware requirements given in the Hardware Requirements table represents a superset of hardware requirements needed for providing Diagnostic coverage for all Failure modes requirements needed for providing Diagnostic coverage for all Failure modes listed in the FMEDA for each device peripheral. If the application is not using a particular peripheral, all the hardware requirements needed for implementing the Diagnostic Mechanisms for that peripheral can be ignored without affecting the overall safety Diagnostic coverage for the application. For example, if the ADC is not being used, then all hardware requirements that are needed for the ADC Diagnostic Mechanisms may be ignored.
- For some Diagnostic Mechanisms where multiple hardware requirements are possible:
  - In some cases, the choice of method to meet the requirement to be used would typically depend on the "free" peripheral resources available for Diagnostic purposes
  - In other cases, the choice of method to meet the requirement may be dictated by the external hardware requirements (and the resultant system cost) of the corresponding Diagnostic methods

⚠ CAUTION **It is the System Integrators' responsibility to verify the compliance of the final application with these assumptions.**

6. Software Requirements on System Level

Important:
Some of the software described may or may not be available from Microchip. Although this manual uses explicit function names, they are mainly intended as placeholders.

The following points should be kept in mind while implementing software requirements for a specific System Integrators application:

- The list of software requirements given below represents a superset of software requirements needed for providing Diagnostic coverage for all Failure modes listed in the FMEDA for each device peripheral. If the application is not using a particular peripheral, all the software requirements needed for implementing the Diagnostic Mechanisms for that peripheral can be ignored without affecting the overall safety Diagnostic coverage for the application. For example, if the ADC is not being used, then all software requirements that are needed for ADC Diagnostic Mechanisms may be ignored.
- For some Failure modes listed in the device FMEDA, multiple Diagnostic Mechanisms are possible. In such cases, the System Integrator is responsible for analyzing the computational requirements (CPU speed, program memory, and data memory) of each Diagnostic Mechanism concerning their respective Diagnostic coverage. This analysis can then be used to select the Diagnostic Mechanisms most suitable for the application, based on factors, such as available computational resources and relative contribution of the specific Failure mode towards the overall safety goals of the item being designed. For example, the following shows two methods that can be used to meet the Safety Goals: FLASH_MEMORY_CHECKSUM_CRC_TEST or FLASH_MEMORY_BLOCK_REPLICATION can be used to verify the Flash memory. In this case, the execution times may vary and should be considered by the System Integrator as a factor in making this selection.
- The required periodicity of executing each Diagnostic test (see “4. Diagnostic Mechanisms” for information on the periodicity of each Diagnostic test) should also be considered in assessing the effectiveness and computational requirements of each software requirement.
  - In general, periodic tests consume more overall CPU bandwidth than on-demand tests, as periodic tests need to be performed even when the corresponding peripheral is not being actively used. Start-up tests have the lowest impact on application performance as these tests are only performed once.
  - The System Integrator is responsible for determining a suitable periodicity (Diagnostic Test Interval) for each software requirement that involves a periodic test as a part of software integration. This determination may be based on the relationship between the Fault Tolerant Time Interval (for Single Point Faults) of the item being designed and the Diagnostic Test Interval for each device peripheral.

⚠ CAUTION **It is the end System Integrators’ responsibility to verify the compliance of the final application with these assumptions.**

Table 6-1. Software Requirements (SWSR)

Requirement ID	Requirement Body
SW_CLOCK_CONTINUOUS_EXTERNAL_MONITOR_01	Application software shall perform CLOCK_CONTINUOUS_EXTERNAL_MONITOR continuously
SW_CLOCK_PERIODIC_MONITOR_01	Application software shall perform CLOCK_PERIODIC_MONITOR periodically
SW_EEPROM_MEMORY_BLOCK_REPLICATION_01	Application software shall perform EEPROM_MEMORY_BLOCK_REPLICATION on demand

.....continued

Requirement ID	Requirement Body
SW_EEPROM_MEMORY_CHECKSUM_CRC_TEST_01	Application software shall perform EEPROM_MEMORY_CHECKSUM_CRC_TEST periodically
SW_FLASH_MEMORY_BLOCK_REPLICATION_01	Application software shall perform FLASH_MEMORY_BLOCK_REPLICATION on demand
SW_FLASH_MEMORY_CHECKSUM_CRC_TEST_01	Application software shall perform FLASH_MEMORY_CHECKSUM_CRC_TEST periodically
SW_FLASH_MEMORY_CRCSCAN_TEST_01	Application software shall perform FLASH_MEMORY_CRCSCAN_TEST both periodically and on start-up
SW_IO_REGISTER_RESET_STATE_CHECK_01	Application software shall perform IO_REGISTER_RESET_STATE_CHECK on start-up
SW_IO_REGISTER_BLOCK_REPLICATION_01	Application software shall perform IO_REGISTER_BLOCK_REPLICATION and test the replicated values periodically
SW_IO_REGISTER_WRITE_READ_TEST_01	Application software shall perform IO_REGISTER_WRITE_READ_TEST on demand
SW_IO_PORTS_CHANGE_NOTIFICATION_TEST_01	Application software shall perform IO_PORTS_CHANGE_NOTIFICATION_TEST periodically
SW_IO_PORTS_INPUT_COMPARISON_01	Application software shall perform IO_PORTS_INPUT_COMPARISON every time it reads the state of an input port pin
SW_IO_PORTS_OUTPUT_MONITOR_01	Application software shall perform IO_PORTS_OUTPUT_MONITOR every time it writes to an output port pin
SW_SRAM_DUPLICATION_01	Application software shall perform SRAM_DUPLICATION on demand
SW_SRAM_MARCH_TEST_01	Application software shall perform SRAM_MARCH_TEST on start-up
SW_STACK_CONTEXT_INTEGRITY_CHECK_01	Application software shall perform a STACK_CONTEXT_INTEGRITY_CHECK on demand
SW_STACK_GUARD_PATTERN_TEST_01	Application software shall perform STACK_GUARD_PATTERN_TEST periodically

7. System Integrator Responsibilities

This section describes the configuration and use of the device that the System Integrator should evaluate for safety applications. The activities carried out during the system and hardware design may include, but are not limited to, the following recommendations. For each recommendation, a qualifier is provided to categorize the item. The qualifiers are:

- **Information:** Defines a property of the device hardware that should be known and fully understood by the System Integrator in order to use the device and/or its functionalities correctly.
- **Recommendation:** A suggestion to the System Integrator for appropriate use of the device. It is not a mandatory item, but the implementation of the “Recommendation” will add some quality (robustness) to the System Integrator’s application.
- **Required:** A must-do, either for the device to work correctly or to leverage some hardware features considered necessary to the implementation of a correct, robust and safe application. These items are also identified by a tag (“AoU-SIR-xx”, where xx is a unique number) as “Assumptions of Use”.
- **Safety: Suggestion:** A suggestion of how the item can be used in the implementation of a Safety Mechanism. A reference to the specific section and paragraph of the standard is provided. This is not intended to be an exhaustive investigation, but just a starting point for the System Integrator’s application implementation.
- **Safety: Redundancy:** Provides guidance and examples of how the item can be used to implement redundancy to improve the application safety level. It is not intended to be an exhaustive description of potential usage.

The System Integrator should always refer to the available device data sheet, and the silicon errata and data sheet clarification document for a more detailed documentation of the various hardware features presented in this section.

8. Metrics

8.1 Failure Rates and FMEDA

Per ISO 26262:2018, FMEDA targets are summarized in the table below. The corresponding FMEDA needs to be completed to assess the quantitative coverage for the Microchip MCU.

Table 8-1. Hardware Architectural Metrics

Hardware Architectural Metrics	MCU Target	System Target ASIL B
Single-Point Fault Metrics	(Note 2)	≥ 90%
Latent Fault Metrics	(Note 2)	≥ 60%
Probabilistic Metric from Random Hardware Failure (PMHF) (Note 1)	$< 10 * 10^{-9}\ h^{-1}$	$< 10^{-7}\ h^{-1}$

Notes:

1. Assumes 10% of total System Target FIT is assigned to the MCU; actual allocation will vary based on system needs.
2. ISO 26262:2018 coverage targets are only specified at the full system level and can vary at the component level.

8.2 IEC 61508: Allowable Safety Integrity Levels

Per IEC 61508-2:2010, the targets are summarized in the table below:

Safe Failure Fraction of an Element	Type A(1)			Type B(2)		
	Hardware Fault Tolerance(3)			Hardware Fault Tolerance(3)		
	0	1	2	0	1	2
< 60%	SIL 1	SIL 2	SIL 3	Not Allowed	SIL 1	SIL 2
60% - < 90%	SIL 2	SIL 3	SIL 4	SIL 1	SIL 2	SIL 3
90% - < 99%	SIL 3	SIL 4	SIL 4	SIL 2	SIL 3	SIL 4
≥ 99%	SIL 3	SIL 4	SIL 4	SIL 3	SIL 4	SIL 4

Notes:

1. Refer to 61508-2 © IEC:2010, Section 7.4.4.1.2, for a definition of a “Type A” element.
2. Refer to 61508-2 © IEC:2010, Section 7.4.4.1.3, for a definition of a “Type B” element.
3. Refer to 61508-2 © IEC:2010, Section 7.4.4.1.1, for a definition of “Hardware Fault Tolerance”.

9. Provisions Against Dependent Failures

This section includes a list of device features that are relevant for the design of robust safety-critical applications.

10. Measures to Prevent Systematic Failures

A systematic error is an error in the software or hardware design/implementation, or of the tools used to implement the application. These errors can be minimized with a strong and precise design flow process. The measures presented in this section provide an additional layer of protection against Systematic Failures.

Some processes implemented by Microchip can help to achieve this target:

- Customer-specific part numbers:
  - Assigned every time a PPAP is issued
  - Ensures that any unique manufacturing or test requirements from the customer are implemented on the specific devices of interest
  - Example: Firmware programming revision control
- Product Change Notification (PCN) subscription service:
  - Customer-specific PCN sent to all customers with a PPAP

11. Assumptions of Use (AoU)

The Microchip was developed according to the Microchip New Product Development standard processes. It can be considered a SEooC since it is a general purpose MCU and has not been designed explicitly for any system.

Microchip provides certain assumptions that shall be performed by the System Integrator to meet technical and functional safety requirements defined at the system level. The Assumptions of Use are described in Sections 11.1-11.5. The details of some of these Assumptions of Use can be found in other sections in this manual, where they are addressed as part of a specific topic being discussed.

It is the responsibility of the System Integrator to address all the Assumptions of Use listed in this manual.

The System Integrator has two options:

- Make sure the assumption is fulfilled
- Disregard the assumption

In both cases, the System Integrator shall provide evidence of the fulfillment and/or a detailed explanation of why disregarding the assumption will not violate a safety requirement or how the assumption has been sufficiently addressed differently.

In the following sections, all the Assumptions of Use relevant to the design of a system using the Microchip devices covered by this manual are listed in summary tables.

12. References

This section is intended to facilitate finding and accessing documents relevant to safety and specific to the device families described in this manual.

Table 12-1. Safety-Relevant Documents

Source	Document	Available from (Links)
International Organization for Standardization	ISO 26262:2018 Standard	www.iso.org/home.html
International Electro-technical Commission (IEC)	IEC 61508, Edition 2.0 2010-04	www.iec.ch
Microchip	Data Sheet	www.microchip.com/
	Silicon Errata and Data Sheet Clarification	
	FMEDA and safety manual	https://www.microchip.com/PIC-AVR-ISO26262 www.microchip.com/PIC-AVR-IEC61508

13. Revision History

Doc. Rev.	Date	Comments
B	05/2022	Added IEC61508 support
A	08/2020	This document is for demonstration purposes only: It describes a non-existing product, and important information is intentionally left out. The document shall not be used as a reference for developing safety applications. Safety manuals for actual products are available on request to Microchip.

Microchip Information

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "unbreakable". Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded

by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, CryptoMemory, CryptoRF, dsPIC, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, Flashtec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet- Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, TrueTime, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, GridTime, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, NVM Express, NVMe, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAM-ICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SmartHLS, SMART-I.S., storClad, SQI, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, TSHARC, USBCheck, VariSense, VectorBlox, VeriPHY, ViewSpan, WiperLock, XpressConnect, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, Symmcom, and Trusted Time are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

ISBN: 978-1-6683-0368-9

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: www.microchip.com/support Web Address: www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4485-5910 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-72400 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820