Document Number: FS6500-FS4500SMUG
Rev. 2.0, 6/2016

Safety Manual for FS6500 and FS4500

Table of Contents

1 Introduction

This document discusses requirements for the use of the FS6500 and FS4500 system basis chip (SBC) family in functional safety relevant applications requiring high functional safety integrity levels. It is intended to support system and software engineers using the FS6500 and FS4500 available features, as well as achieving additional diagnostic coverage by software measures.

Several measures are prescribed as safety requirements whereby the measure described was assumed to be in place when analyzing the functional safety of this system basis chip. In this sense, requirements in the safety manual are driven by assumptions **[SMR_xx]** concerning the functional safety of the system integrating the FS6500 and FS4500.

- **Assumption**: An assumption being relevant for functional safety in the specific application under consideration (condition of use). It is assumed the user fulfills an assumption in his design.

For the use of the system basis chip, means if a specific safety manual assumption is not fulfilled, it has to be rationalized an alternative implementation is at least similarly efficient concerning the functional safety requirement in question similarly well (for example, provides same coverage, reduces the likelihood of common mode failure (CMF), and so on), or the estimation of an increased failure rate (λ_{SPF}, λ_{RF}, λ_{MPF}, λ_{DU}…) and reduced metrics (SFF: safe failure fraction, SPFM: single-point fault metrics, LFM: latent fault metric) due to the deviation has be specified.

This document also contains guidelines on how to configure and operate the FS6500 and FS4500 for functional safety relevant applications requiring high functional safety integrity levels. These guidelines are preceded by one of the following text statements:

- **Recommendation**: A recommendation is either a proposal for the implementation of an assumption, or a reasonable measure which is recommended to be applied, if there is no assumption in place. The user has the choice whether to follow the recommendation.
- **Rationale**: The motivation for a specific assumption and/or recommendation.
- **Implementation hint:** An implementation hint gives specific hints on the implementation of an assumption and/or recommendation on the FS6500 and FS4500. The user has the choice whether to follow the implementation hint.

These guidelines are considered to be useful approaches for the specific topics under discussion. The user needs to use discretion in deciding whether these measures are appropriate for their applications.

This document is valid only under the assumption the system basis chip is used in functional safety applications requiring a fail-silent or a fail-indicate system basis chip. A fail-operational mode of the FS6500 and FS4500 is not described.

This document targets high functional safety integrity levels. For functional safety goals which do not require high functional safety integrity levels, system integrators need to tailor the requirements for their specific application.

It is assumed, the user of this document is generally familiar with the FS6500 and FS4500 device and ISO 26262 standard.

1.1 Certified ASG NPI flow

FS6500 and FS4500 follows SafeAssure company wide analog & sensors hardware development process for functional safety related components.

The audited SafeAssure development process for hardware product components complies with the following ISO 26262 standard part requirements:

- ISO 26262-2:2011
- ISO 26262-5:2011
- ISO 26262-7:2011
- ISO 26262-8:2011
- ISO 26262-9:2011

1.2 Tailored life cycle description

The FS6500 and FS4500 is not an item in the sense of ISO 26262, but only is a component in an item developed at a later point in time. Therefore, the FS6500 and FS4500 project follows a tailored “safety element out of context” (SEooC) life cycle.

Along with the hardware part, usual documentation (data sheet including electrical parameters) and safety related documentation (safety manual, FMEDA) are provided to parties which integrate the FS6500 and FS4500 into systems.

The following table gives detail of the specific tailoring of the safety life cycle applicable for the FS6500 and FS4500 development.

Table 1. Tailoring details

ISO26262 part	ISO26262 section	Topic of the part	Applicability	Justification or exceptions
1	All sections	Vocabulary	Applicable	-
2	All sections	Management of functional safety	Applicable	-
3	All sections	Concept phase	NOT Applicable	Under customer responsibility
4	All sections	Product development at system level	NOT Applicable	Under customer responsibility
5	All sections	Product development at hardware level	Applicable	-
6	All sections	Product development at software level	NOT Applicable	Under customer responsibility
7	All sections	Production and operation	Applicable	No maintenance, no reparation and no decommissioning planned at product level. The maintenance and reparation can be done only at system or vehicle level
8	All sections	Supporting processes	Applicable	There is no distributed development on the FS6500 and FS4500 development. Qualification of software component: software and the reuse of software components was not part of FS6500 and FS4500 development. No proven in use argument is considered in the context of FS6500 and FS4500 development
9	All sections	ASIL-oriented and safety -oriented analyses	Applicable	There is no ASIL decomposition to consider in the FS6500 and FS4500 development.
10	All sections	Guideline on ISO 26262	NOT Applicable	Information part only

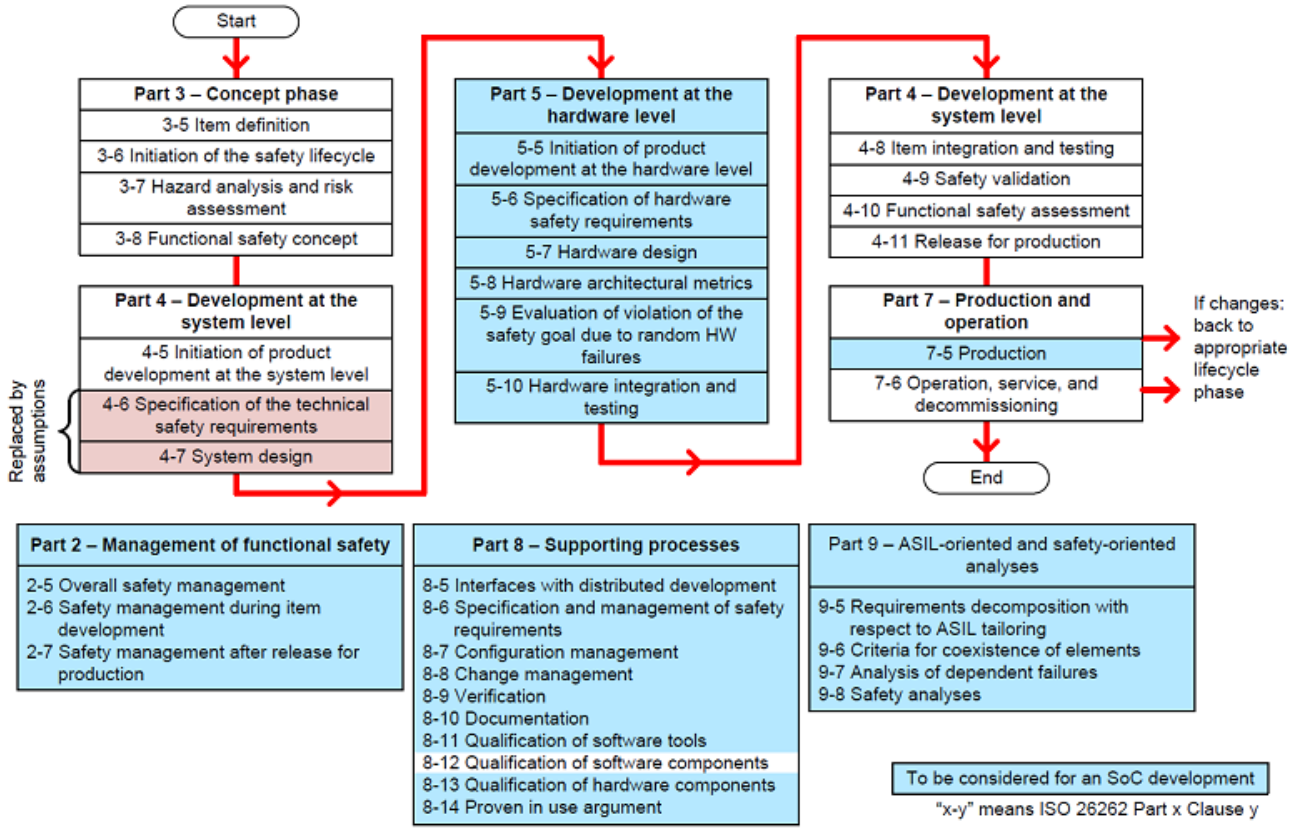


Figure 1. Applicable ISO26262 parts for FS6500 and FS4500 development

1.3 Customer task responsibility

In a context of customer applications, this is a list of required customer tasks under their responsibility. The list is delivered as an example and is not exhaustive. In case of questions, the customer should contact their local NXP representative.

- Use of the latest FS6500 and FS4500 documentation revision (data sheet, safety manual, FMEDA, application notes, errata,...) available at www.nxp.com.
- Other or additional safety requirements might have to be considered depending of the target application and required standard (e.g. IEC 61508, IEC 61784, etc).
- Verify the application mission profile is well covered by the FS6500 and FS4500 devices as showed in Table 2 of 2.2, Operation of use and mission profile, page 9.
- Compare system requirements versus FS6500 and FS4500 requirements and make sure there are no deviances.
- Establish validity of assumptions at the system level considered in 2, Assumptions of use, page 9.
  - Verify the fault tolerant time interval of the FS6500 and FS4500 is under the system FTTI requirement, whatever the faults
  - Verify the violation of the technical assumptions as described in 2.6, Technical safety requirements, page 13.
  - Verify the safe state considerations described in 2.4, Safe state, page 11.
- Perform safety analysis at the system level, taking into account the safety analysis provided for the FS6500 and FS4500. Consider assumptions like typical mission profile and failure rate data book (IEC 62380).
- During safety analysis, the non-functional blocks (e.g. debug, etc) should also be considered.
- Perform calculation and verify the safety metrics.
- Perform DFA analysis.
- Validate FS6500 and FS4500 outputs behave as expected in the application, and also during of an error condition.
- Consider and verify single point failures and latent failures at system level.
- Consider and verify systematic errors during development.
- Verify the effectiveness of diagnostics at the system level
- Perform fault injection tests and validate safety mechanisms.
- Consider all recommendations and implementation hints given in this safety manual.
- For completeness of ISO26262 compliance at system level, consider the 1.2, Tailored life cycle description, page 3, Table 1.
- The installation of the device at the module level is the responsibility of the customer. However, NXP gives recommendations on NXP QFP packages during printed circuit board (PCB) assembly. This document serves only as a guideline to help users develop a specific solution. Actual experience and development efforts are still required to optimize the assembly process and application design per individual device requirements, industry standards such as IPC and JEDEC, and prevalent practices in the user's assembly environment.

1.4 Safety documentation set

This sections lists all the helpful documentation for the user.

Document number	Document type	Description
ISO 26262	Standard	ISO 26262 Road vehicles - Functional safety, November 2011
IEC TR 62380	Standard	Reliability data handbook – Universal model for reliability prediction of electronics components, PCBs and equipment
FCTNLSFTYWP	White paper	Addressing the Challenges of Functional Safety in the Automotive and Industrial Markets, White Paper, October 2011
FS6500-FS4500	Data Sheet	Data Sheet for FS6500 & FS4500
doc_FS6500_FS1B _Dynamic_FMEDA	Dynamic FMEDA	Failure Mode Effects and Diagnostic Analysis Document.
FS/71/220/15/0078	TUV SAAR Certificate	Conformity of development and support process in accordance with ISO26262
201618041A	PPAP	Report summarizing data gathered during qualification of the FS6500 and FS4500 following AECQ100-RevG requirements.
	Report	ISO26262 ASIL-D assessment report

1.5 Vocabulary

For the purposes of this document, the vocabulary defined in ISO 26262-1 apply to this document.

Specifically, the following terms apply.

- **System:** functional safety-related system implementing the required functional safety goals necessary to achieve or maintain a safe $state_{system}$ for the equipment under control (control system). The system is intended to achieve on its own or with other electrical/electronic/programmable electronic functional safety-related systems, the necessary functional safety integrity for the required safety functions.
- **System integrator:** the person responsible for the system integration.
- **Element:** part of a subsystem comprising of a single component or any group of components (for example, hardware, software, hardware parts, software units) performing one or more element safety functions (functional safety requirements).
- **Trip time:** the maximum time of operation of the SBC without switching to a power down state.

1.6 Faults and failures definition

Failures are the main impairment to functional safety:

- A **systematic failure** is manifested in a deterministic way to a certain cause (systematic fault), which can only be eliminated by a change of the design process, manufacturing process, operational procedures, documentation, or other relevant factors. Thus, measures against systematic faults are reductions of systematic faults, for example, implementing and following adequate processes.
- A **random hardware failure** can occur unpredictably during the lifetime of a hardware element and follows a probability distribution. Thus, measures reducing the likelihood of random hardware faults are either the detection and control of the faults during the lifetime, or reduction of failure rates. A random hardware failure is caused by either a permanent fault (for example, physical damage), an intermittent fault, or a transient fault. Permanent faults are unrecoverable. Intermittent faults are for example, faults linked to specific operating conditions or noise. Transient faults are for example, EMI-radiation. An affected configuration register can be recovered by setting the desired value or by a power cycle. Due to a transient fault, an element may be switched into a self-destructive state (for example, single event latch up), and therefore may cause permanent destruction.

1.6.1 Faults

The following random faults may generate failures, which may lead to the violation of a functional safety goal. Citations are according to ISO 26262-1. Random hardware faults occur at a random time, which results from one or more of the possible degradation mechanisms in the hardware.

- **Single-point fault (SPF):**
  An SPF is ‘a fault in an element not covered by a safety mechanism’ and results to a single-point failure ‘which leads directly to the violation of a safety goal’. Figure 2a shows an SPF inside an element generating a wrong output.
- **Latent fault (LF):**
  An LF is a ‘multiple-point fault whose presence is not detected by a safety mechanism nor perceived by the driver’. An LF is a fault which does not violate the functional safety goal(s) itself, but leads in combination with at least one additional independent fault to a dual- or multiple-point failure, which then leads directly to the violation of a functional safety goal. Figure 2b shows an LF inside an element, which still generates a correct output.
- **Residual fault (RF):**
  An RF is a ‘portion of a fault which by itself leads to the violation of a safety goal’, ‘where the portion of the fault is not covered by a functional safety mechanism’. Figure 2c shows an RF inside an element, which - although a functional safety mechanism is set in place - generates a wrong output, as this particular fault is not covered by the functional safety mechanism.
- **Dual-point fault (DPF):**
  A DPF is an ‘individual fault which, in combination with another independent fault, leads to a dual-point failure’, which leads directly to the violation to a goal. Figure 2d shows two LF inside an element generating a wrong output.
- **Multiple-point fault (MPF):**
  An MPF is an ‘individual fault which, in combination with other independent faults, leads to a multiple-point failure’, which leads directly to the violation of a functional safety goal. Multiple-point faults are not covered in functional safety concept of the FS6500 and FS4500.
- **Safe Fault (SF):**
  An SF is a ‘fault whose occurrence does not significantly increase the probability of violation of a safety goal’. Safe faults are not covered in this document. Single-point faults, residual faults, or dual-point faults are not safe faults.

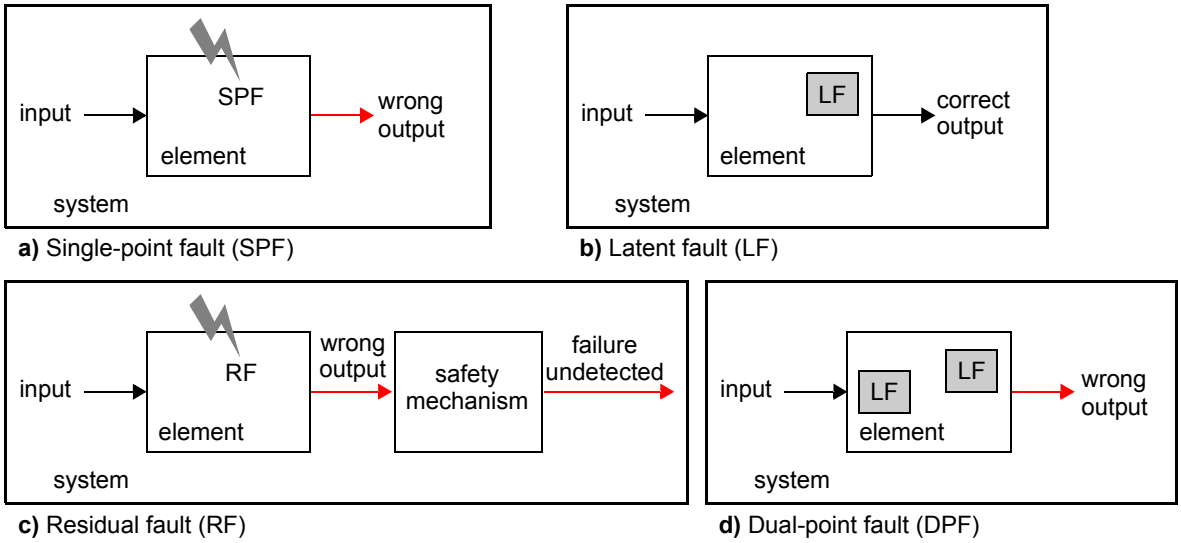


Figure 2. Faults

SPFs must be detected within the FTTI. Latent faults (dual-point faults) must be detected within the MPFDI. In automotive applications, MPFDI is generally accepted to be once per typical automotive trip time (t_{TRIP}) by test routines (for example, BIST after power-up). This reduces the accumulation time of latent faults from the lifetime of the product t_{LIFE} to t_{TRIP}.

2.2, Operation of use and mission profile, page 9 lists a profile with a typical trip time for automotive applications.

1.6.2 Failures

- **Common cause failure (CCF):**
  CCF is a coincidence of random failure states of two or more elements in separate channels of a redundancy element, leading to the defined element failing to perform its intended safety function, resulting from a single event or root cause (chance cause, non-assignable cause, noise, natural pattern, …). common cause failure causes the probability of multiple channels (N) having a failure rate to be larger than $\lambda_{\text{single channel}}^{N}$ ($\lambda_{\text{redundant element}} > \lambda_{\text{single channel}}^{N}$).

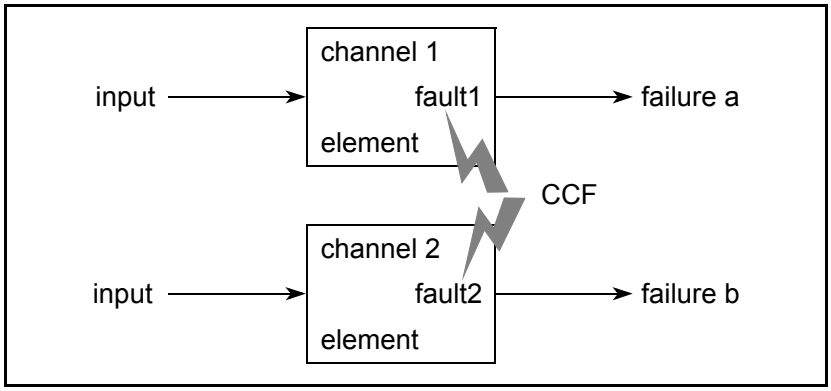


Figure 3. Common cause failures

- **Common mode failure (CMF):**
  CMF is a subset of CCF. A single root cause leads to similar coincidental erroneous behavior (with respect to the safety function) of two or more (not necessarily identical) elements in redundant channels, resulting in the inability to detect the failures. Figure 4 shows three elements within two redundant channels. One single root cause (CMF A or CMF B) leads to undetected failures in the primary channel and in one of the elements of the redundant channel.

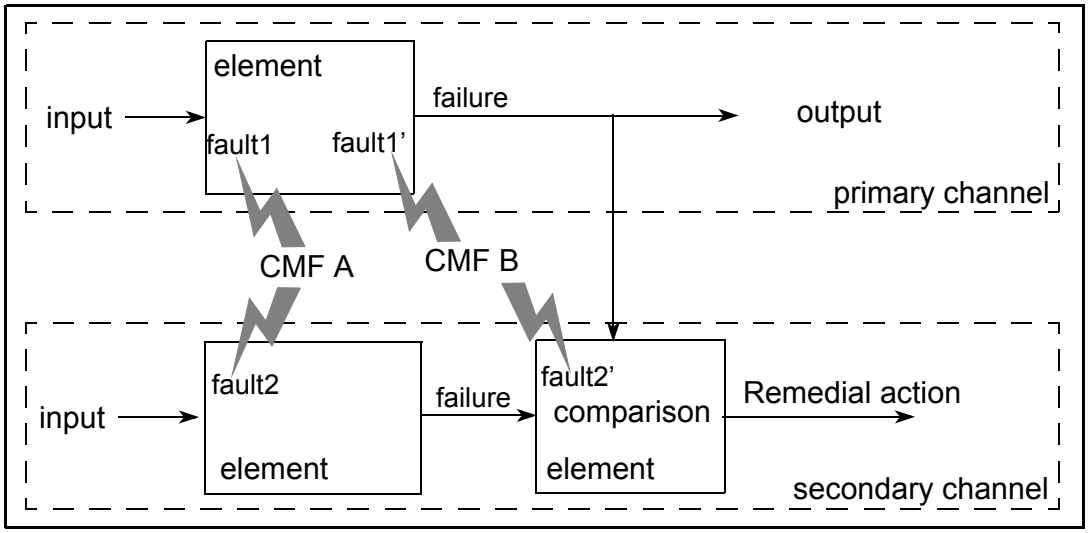


Figure 4. Common mode failures

- **Cascaded failure (CF):**
  CFs occur when local faults of an element in a system ripple through interconnected elements causing another element or elements of the same system and within the same channel to fail. Cascading failures are dependent failures, not common cause failures. Figure 5 shows two elements within a single channel, to which a single root cause leads to a fault (fault 1) in one element resulting in a failure (failure a), and causing a second fault (fault 2) within the second element (failure b).

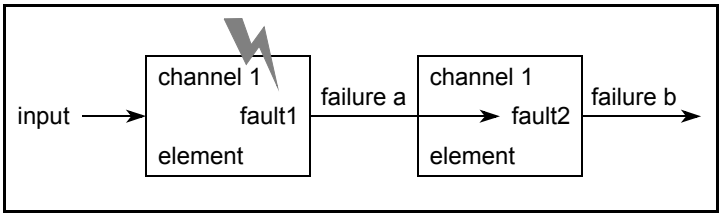


Figure 5. Cascading failures

2 Assumptions of use

The part numbers supported by this safety manual are listed in the FS6500-FS4500 data sheet document.

2.1 Generic safety system architecture

The FS6500 and FS4500 are designed to be used in automotive or industrial applications which are needed to fulfill functional safety requirements, as defined by functional safety integrity levels (for example, ASIL D of ISO 26262).

The Figure 6 shows a generic safety system architecture example. The FS6500 and FS4500 are intended to be the main power supply for the MCU (V_{CORE} and V_{CCA}) and the sensor (V_{AUX}), with MCU monitoring (watchdog and FCCU) and Fail-safe outputs (FS0B and FS1B) to put the system in safe-state.

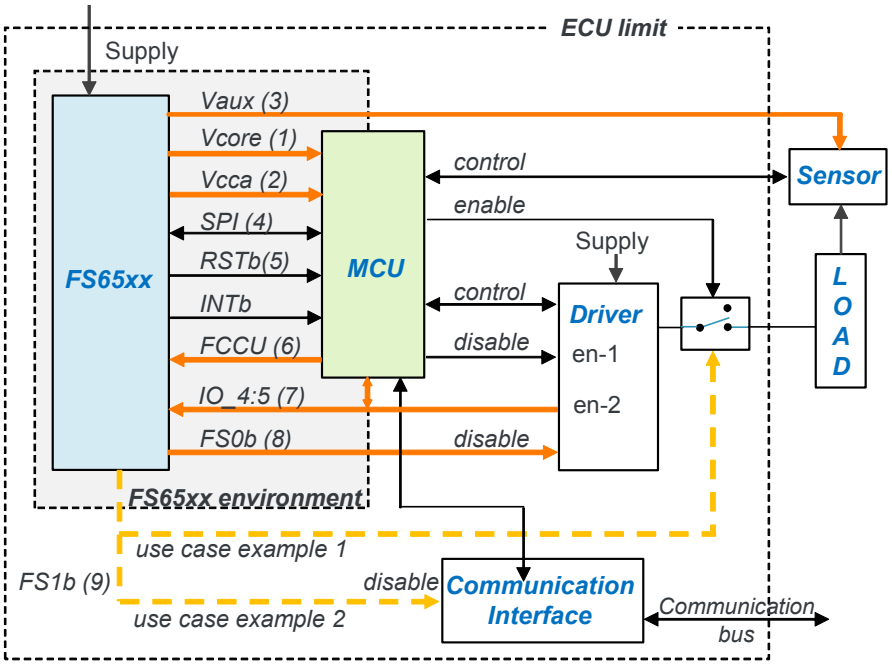


Figure 6. Generic safety system architecture

- V_{CORE} (1): MCU core supply (safety related function)
- V_{CCA} (2): MCU ADC reference voltage and/or input/output supply (safety related function)
- V_{AUX} (3): Sensor or local supply (safety related function)
- SPI (4): Serial peripheral Interface between MCU and FS6500 or FS4500 used for watchdog (safety monitoring)
- RSTB (5): Active low bidirectional reset (not safety)
- FCCU (6): FCCU input monitoring (safety monitoring)
- IO_4:5 (7): External IC error monitoring (safety monitoring)
- FS0B (8): Primary Fail-safe output (safety related function)
- FS1B (9): Secondary Fail-safe output (safety related function)

2.2 Operation of use and mission profile

The FS6500 and FS4500 are developed to target the highest level of safety integrity (ASIL D) when applying all recommendations and applicability of the system assumptions mentioned in this safety manual, and for the mission profile of typical safety automotive applications.

Assumption: [SMR_01] It is assumed the FS6500 and FS4500 are used in '12 volts automotive' applications for which the battery voltage (i.e. pin VSUP1, VSUP,2, VSUP3, and VSENSE) never exceeds the maximum ratings of the FS6500 and FS4500 (i.e 40V). Above this voltage, the FS6500 and FS4500 run the risk of being destroyed and the safety requirements are no longer satisfied.[END]

Assumption: [SMR_02] It is assumed the FS6500 and FS4500 are used in applications for which the fault tolerant time interval is ≥ 10 ms. Shorter 'fault tolerant time interval' must be deeply analyzed.[END]

Assumption: [SMR_03] It is assumed the FS6500 and FS4500 are used in applications for which the mission profile is equivalent to or less aggressive, Table 3., Temperature profile for mission profiles, page 10.[END]

Assumption: [SMR_04] It is assumed when the multiple point fault time interval is ≤ 12 hours, then the driving cycle is assumed to be ≤ 12 hours.[END]

Assumption: [SMR_05] It is assumed the normal operating range of the FS6500 and FS4500 is fulfilled by the compliance to the FS6500 and FS4500 data sheet.[END]

Assumption: [SMR_06] To avoid systematic errors during system integration, It is system integrator's responsibility to follow NXP recommendations as described in the FS6500 and FS4500 data sheet and application note available at www.nxp.com.[END]

Assumption: [SMR_07] It is system integrator's responsibility to report all field failures of the devices to the silicon supplier.[END]

Rationale: To cover the ISO 26262-7 (6.5.4) and ISO 26262-7 (6.4.2.1).

Assumption: [SMR_08] It is system integrator's responsibility to take into account the latest device errata during system design, implementation, and maintenance. For a functional safety-related device such as FS6500 and FS4500, this also concerns functional safety-related activities such as system level functional safety concept development.[END]

The FS6500 is used in applications for which the mission profile is described in Table 2. This document is based on these mission profiles, although use of the FS6500 and FS4500 are not limited to these values. Mission profile is a typical automotive profile.

To prevent latent fault accumulation during a trip time longer than 12 hours, additional diagnostic measures need to be executed within the additional multiple-point fault detection interval (MPFDI).

Table 2. Mission profiles

Mission parameters	Mission profile
Junction temperature	-40 °C to 150 °C
Trip time (t_{TRIP})	12 hours
FTTI	10 ms
Lifetime (t_{LIFE})	20 years
Total operating hours	12000 hours

Table 3 shows temperature profiles.

Table 3. Temperature profile for mission profiles

Device type	Temperature range (°C)	Operation time (h)
Packaged device	125	120
	120	960
	76	7800
	23	2400
	-40	720

Thermal cycling considered:

- 335 operating days with four day light starts with temperature increase of 62 °C.
- 335 operating days with two night starts with temperature increase of 72 °C.
- 30 days non-operating with temp increase of 10 °C.

2.3 Safety integrated level

Table 4. Safety integrated level

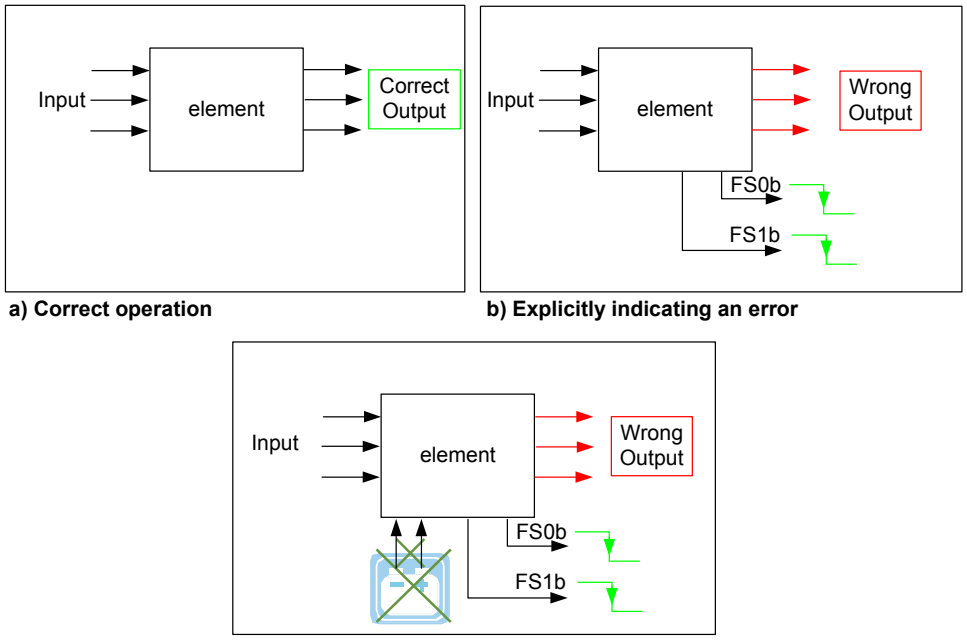
Safety related function	ASIL level
The FS6500 and FS4500 must provide V_{CORE}, V_{CCA}, V_{AUX} supplies within the specified voltage range	ASIL D
The FS6500 and FS4500 must perform a periodic handshake (watchdog) in order to confirm the correct behavior of the MCU	ASIL D
The FS6500 and FS4500 must listen to the FCCU error signal of the MCU	ASIL D
The FS6500 and FS4500 must listen to an external IC error signal (other than MCU)	ASIL D
The FS6500 and FS4500 must indicate a Fail-safe state by virtue of the FS0B output within the FTTI time	ASIL D
The FS6500 and FS4500 must indicate a Fail-safe state by virtue of the FS1B output after a delay or for a duration when FS0B is activated	ASIL D

2.4 Safe state

A safe state of the system is named safe $state_{system}$ whereas a safe state of the FS6500 and FS4500 is named safe $state_{SBC}$. A safe $state_{system}$ of a system is an operating mode without an unreasonable probability of occurrence of physical injury or damage to the health of persons. A safe $state_{system}$ may be the intended operating mode or a mode where the system has been disabled.

Likewise, a safe $state_{SBC}$ of the FS6500 and FS4500 is by definition one of following operation modes (see Figure 7):

a) Operating correctly
— Outputs depend on application
b) Explicitly indicating an error (safe $state_{SBC}$)
— Fail-safe outputs FS0B and FS1B indicating an error (active-low)
c) Completely unpowered



a) Correct operation

b) Explicitly indicating an error

c) Completly unpowered

Figure 7. Safe $state_{SBC}$ of the FS6500 and FS4500

Assumption: [SMR_09] It is the system integrator's responsibility to ensure the system transitions itself to a safe $state_{system}$ when the FS6500 and FS4500 explicitly indicates an error via its Fail-safe outputs (FS0B and FS1B).[END]

Assumption: [SMR_10] It is the system integrator's responsibility to ensure the system transitions itself to a safe $state_{system}$ when the FS6500 and FS4500 are completely unpowered.[END]

2.5 Single-point fault tolerant time interval and process safety time

The single-point fault tolerant time interval (FTTI)/process safety time (PST) is the time span between a failure having the potential to give rise to a hazardous event, and the time by which counteraction has to be completed to prevent the hazardous event from occurring. It is used to define the sum of the worse case fault indication time and the time for execution of corresponding countermeasures (reaction). Figure 8 shows the FTTI for a single-point fault occurring in the SBC (Figure 8a) with an appropriate functional safety mechanism to handle the fault (Figure 8b). Without any suitable functional safety mechanism, a hazard may appear after the FTTI elapsed (Figure 8c).

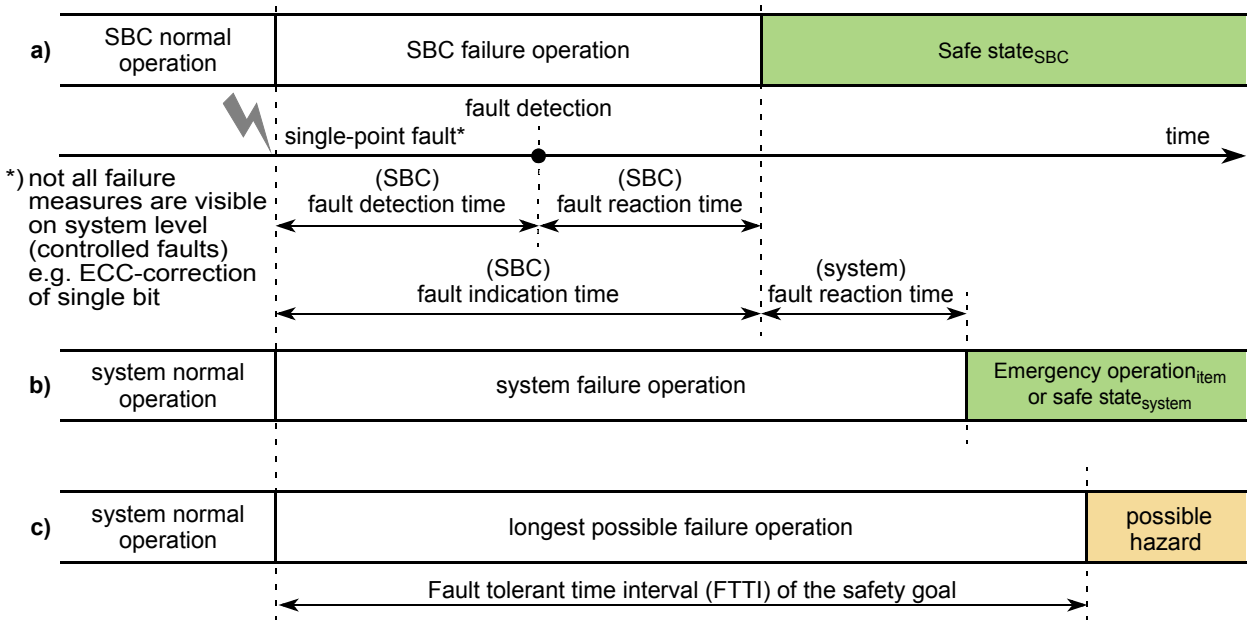


Figure 8. Fault tolerant time interval for single-point faults

Fault indication time is the time it takes from the occurrence of a fault to switching into safe state$_{SBC}$ (for example, indication of the failure by asserting the Fail-safe output pins).

Fault indication time of the SBC has three components:

Fault indication time = **recognition time** + **internal processing time** + **external indication time**.

Each component of fault indication time is described as follows:

- **Fault detection time** is the maximum time for detection of a fault and consists of:
  - **Recognition time** is the maximum of the recognition time of all involved functional safety mechanisms. The two mechanisms with the longest time are:
    - Recognition time of an overvoltage on regulators takes 234 µs maximum (corresponding to filtering time)
    - Acknowledgement counter used for external IC monitoring is maximum 9.7 ms
  - **Fault reaction time** is the maximum of the reaction time of all involved functional safety mechanisms consisting of internal processing time and external indication time:
  - **Internal processing time** is not the same depending of the origin but the longest time is about 80 µs for an overvoltage detection. All others are below 80 µs.
  - **External indication time** to notify an observer about the failure external to the SBC. This time is 3.0 µs for RSTB, 22 µs for FS0B and FS1B when used in t_{DUR} configuration. Time needed to activate Fail-safe outputs when the internal command is sent from digital and activates the analog drivers. This time is in addition to configurable t_{DELAY} for FS1B when used in t_{DELAY} configuration (refer to the FS6500 and FS4500 data sheet for t_{DELAY} timings).

The sum of the SBC fault indication time and system fault reaction time must be less than the FTTI of the functional safety goal, except when FS1B is used in t_{DELAY} configuration with t_{DELAY} > FTTI (5.7.2, FS1B, page 41).

2.6 Technical safety requirements

List of the assumed technical safety requirements **[TSR_xx]** of the element considered as potentially violating the safety goals.

Assumption: [TSR_01] It is assumed the FS6500 and FS4500 are used in combination with other devices in the application (i.e. MCU, other analog IC).[END]

Assumption: [TSR_02] It is assumed an out of range operation of the power management integrated circuit (V_{CORE}, V_{AUX}, and V_{CCA}) is considered a violation of at least one of the safety goals of the system.[END]

Assumption: [TSR_03] It is assumed the FS6500 and FS4500 provide the safety mechanism for undervoltage and overvoltage monitoring of the power management integrated circuit. When the monitoring detects a fault, it activates the safe $state_{SBC}$ within the FTTI.[END]

Assumption: [TSR_04] Faults having a direct impact on violation of TSR03 are assumed as single point faults.[END]

Assumption: [TSR_05] It is assumed an abnormal SW & HW execution of the MCU is considered a violation of at least one of the safety goals of the system.[END]

Assumption: [TSR_06] It is assumed the FS6500 and FS4500 provide the safety mechanism for temporal and logical monitoring (watchdog) of an MCU. When the monitoring detects a fault, it activates the safe $state_{SBC}$ within the FTTI.[END]

Assumption: [TSR_07] Faults having a direct impact on the violation of TSR06 are assumed as latent faults.[END]

Assumption: [TSR_08] It is assumed the FS6500 and FS4500 provide the safety mechanism for MCU error monitoring (FCCU). When the monitoring detects a fault, it activates the safe $state_{SBC}$ within the FTTI.[END]

Assumption: [TSR_09] Faults having a direct impact on a violation of TSR08 are assumed as latent faults.[END]

Assumption: [TSR_10] It is assumed the FS6500 and FS4500 provide the safety mechanism for an external IC(s) error monitoring (IO_4:5). When the monitoring detects a fault, it activates the safe $state_{SBC}$ within the FTTI.[END]

Assumption: [TSR_11] Faults having a direct impact on a violation of TSR10 are assumed as latent faults.[END]

Assumption: [TSR_12] It is assumed a self-test (LBIST/ABIST) during start-up is performed to ensure the integrity of the system and to prevent latent faults. In case of a latent fault, the application stays in the safe state.[END]

3 Failure rates and FMEDA

3.1 Failure handling

Failure handling can be split into two categories:

- Failure handling before enabling the system level safety function (for example, during/following the MCU initialization). These errors are required to be handled before the system enables the safety function, or in a time shorter than the respective FTTI after enabling the safety function.
- Failure handling during runtime with repetitive supervision while the safety function is enabled. These errors have to be handled in a time shorter than the respective FTTI.

Assumption: [SMR_11] It is assumed single-point and latent fault diagnostic measures complete operations (including fault reaction) in a time shorter than the respective FTTI when the safety function is enabled.[END]

Recommendation: It is recommended to identify startup failures before enabling system level safety functions.

A typical failure reaction regarding power-up/start-up diagnostic measures is not to initialize and start the safety function, but instead to provide failure indication to the operator/user.

3.2 Failure rates

The FS6500 and FS4500 failure rate data is derived from the IEC/TR 62380, to quantify the hardware architectural metrics for the evaluation of the effectiveness of the design architecture against the requirements for random hardware failures handling.

The random hardware failures addressed by these metrics are limited to some of the item's safety-related electrical and electronic hardware parts, namely those which can significantly contribute to the violation or the achievement of the safety goal, and to the single-point, residual, and latent faults for those parts. Only the electrical failure modes and failure rates are considered for the FS6500 and FS4500.

The IEC/TR 62380 considers the failure rate model for permanent faults in a semiconductor device to be the sum of three subcomponents:

- the **Die** predictive failure rate (10.02 FIT), which depends mainly on
  - silicon parameters like the technology and its maturity, the number of transistors
  - and application parameters like the mission profile, the power dissipation and the junction to ambient thermal resistance
- the **Package** predictive failure rate (36.12 FIT), which depends mainly on
  - package parameters like the number of pins, the pitch
  - and application parameters like the mission profile, the temperature cycles, the board material
- the **Interface electrical overstress** predictive failure rate (7.46 FIT), which depends mainly on
  - application parameters like cable length attached to global pins (three meters considered)

with a total device failure rate of **53.6 FIT**.

The transient faults are not considered for FS6500 and FS4500 developed in 0.25 µm technology without SRAM. The only potential concern for soft errors would be in logic latches and flops. Logic soft error upsets are simply not a significant risk at this size nodes, due to higher voltage and capacitance, making it very difficult to cause an upset by radiation. However, advanced digital architecture mechanisms have been implemented into the FS6500 and FS4500 devices to detect permanent faults, and some can also mitigate transient fault. Those technics are the redundancy of the state machine, the oscillator, filtering glitches, Hamming, and ECC technics.

The method used to evaluate and to quantify the hardware architectural metrics is based on the FMEDA, which details the determination of error causes and their impact on the system. The hardware architectural metrics are dependent upon the context of use of the FS6500 and FS4500:

- Mission profile of the application in which the FS6500 and FS4500 are operating
- Selection/usage of the functions and functional safety mechanisms implemented in the application

3.3 FMEDA overview

The FS6500 and FS4500 are developed according to the ISO26262 standard, then a functional safety failure analysis on the hardware design was performed to identify failure causes and their effects and quantitative safety metric values.

FMEDA inductive analysis as the method was applied. This FMEDA is based on microsoft excel sheets with the capability to enable safety analysis of the FS6500 and FS4500 features implemented for a specific application.

The FS6500 and FS4500 FMEDA sheet is an example only, based on the result of the safety analysis performed for the context of use of the FS6500 and FS4500, using the mission profile described in Table 3., Temperature profile for mission profiles, page 10, and when applying all recommendations and assumptions mentioned in this safety manual.

Assumption: [SMR_12] It is assumed simultaneous pin disconnections (i.e. pin lift on the PCB) are restricted to 1 during pin FMEA and FMEDA analysis.[END]

Assumption: [SMR_13] It is assumed the thermal connection of the exposed pad to the PCB is always ensured due to its large size.[END]

Assumption: [SMR_14] Short-circuit between PCB tracks is not considered.[END]

Assumption: [SMR_15] External component disconnection is not considered.[END]

In a context of customer applications, the FMEDA example provided by NXP must be customized to fit for the application requirements. The final customized FMEDA is under the responsibility of the customer, and then solely responsible for the safety metric values.

The FS6500 and FS4500 FMEDA document associated with the FS6500 and FS4500 failure rate estimation document is available upon request, when covered by a NXP Semiconductors NDA (contact your local NXP representative).

NOTE

NXP SafeAssure solutions, including hardware, software and tools help customers design end products which may be used in applications such as safety-critical systems, other than life support and safety critical medical applications. For safety-critical systems buyers and/or customers agree they have all competencies and expertise to design safety critical function(s) to ensure system failures are identified, are anticipated, are monitored, and are controlled following the recommendations and the requirements of the functional safety standards. Buyers and/or customers are solely responsible to meet and comply with applicable functional safety, regulatory standards and safety-related requirements concerning their systems and end-product solutions.

4 Functional safety concept

4.1 Hardware requirements at the system level

This section lists necessary or recommended measures on the system level for the FS6500 and FS4500 to achieve the functional system safety goal(s).

The FS6500 and FS4500 offer an integrated functional safety architecture, a variety of replicated function blocks, self-test unit, and other items to detect faults. By these means, single point failures and latent failure can be detected with a high diagnostic coverage.

However, not all failure modes may be detected on a complete system by the FS6500 and FS4500, so it is assumed a separate circuitry is used to bring the system into the safe $state_{system}$ (MCU) in such cases.

Figure 9 depicts the functional safety related elements of the FS6500 and FS4500.

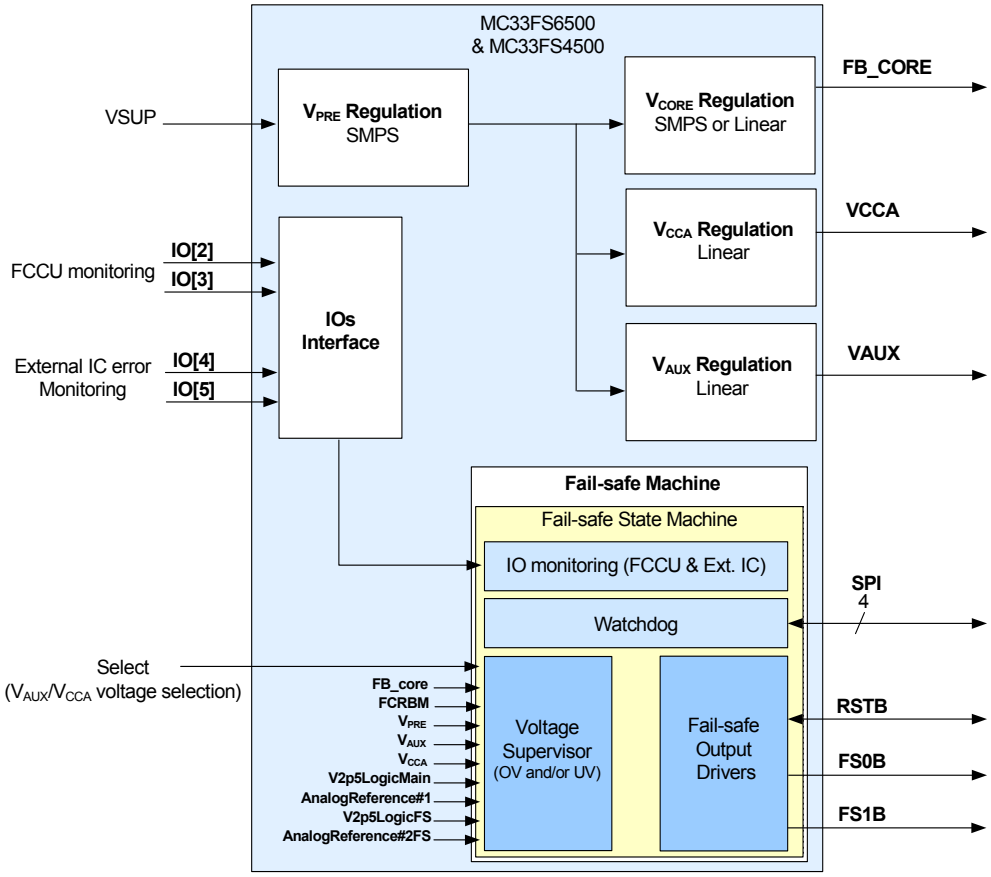


Figure 9. FS6500 and FS4500 functional safety blocks

- V_{PRE} is a current mode SMPS regulator supplying several blocks inside the FS6500 and FS4500, like internal reference voltages, linear regulators, and SMPS regulators.
- V_{CORE} is a voltage mode SMPS regulator for the FS6500 and a linear voltage regulator for the FS4500. It is dedicated to the MCU core supply (from 1.0 V to 5.0 V), configurable through an external resistor bridge. The FCRBM pin monitors the external resistor bridge through a redundant path.
- V_{CCA} is a 5.0 V/3.3 V linear voltage regulator dedicated usually to the MCU ADC reference voltage.
- V_{AUX} is a linear voltage regulator usually dedicated to auxiliary functions (3.3 V/5.0 V) or a sensor supply (tracking of V_{CCA}).
- The SELECT pin is used to configure the voltage level of the V_{CCA} and V_{AUX} regulators, and enable or disable the deep fail safe feature.
- Based on safety requirements, the IOs can be used to monitor external error signals coming from the MCU or from other integrated circuits in the system.

- The Fail-safe machine (FSM) is part of the safety system partitioning. This FSM is made of four main blocks which are:
  - Voltage supervisor (VS)
  - Fail-safe output drivers (FSO)
  - Watchdog (WD)
  - IO monitoring (IO_2:3 for FCCU and IO_4:5 for external IC error)

Figure 10 depicts a simplified application schematic for a functional safety relevant application in conjunction with an MCU (only functional safety-related elements shown). The FS6500 and FS4500 supply the MCU with the required supply voltages (from 1.0 V to 5.0 V). Voltages generated by the FS6500 and FS4500 are monitored for overvoltage and undervoltage by the embedded voltage supervision.

The FS6500 and FS4500 also monitor the state of the error out signals FCCU_F[n] (error monitor) using bi-stable protocol only.

Via the SPI communication interface, the FS6500 and FS4500 repetitively trigger the watchdog from the MCU with a valid answer. A dedicated Fail-safe state machine is implemented to bring and maintain the application in safe $state_{system}$. During a failure (e.g. V_{CORE} overvoltage or undervoltage), RSTB and/or FS0B (depending on the device configuration done during INIT_FS phase) are asserted low.

- The reset pin (RTSB) of the FS6500/FS4500 controls and monitors the reset pin of the MCU.
- A Fail-safe output (FS0B) is available to control or deactivate any Fail-safe circuitry (e.g. power switch) in redundancy with the MCU (red connection) or with FS1B (blue connection).
- A Fail-safe output (FS1B) is available to control or deactivate any Fail-safe circuitry with a delay or for a duration when FS0B is activated. FS1B can be used in redundancy of FS0B, activated at the same time as the FS0B when the delay is configured at 0.

An interrupt output (INTB) for error information is connected to the NMI input of the MCU.

By a connection of the signal MUX_OUT to an ADC-input of MCU, further diagnostic measures are possible (e.g. reading temperature or measuring V_{BATT}). Digital inputs (IO_4, IO_5) may be used for monitoring error signal handling of other devices. Additionally, the FS6500 and FS4500 may act as a physical interface to connect the MCU directly with a CAN or LIN bus.

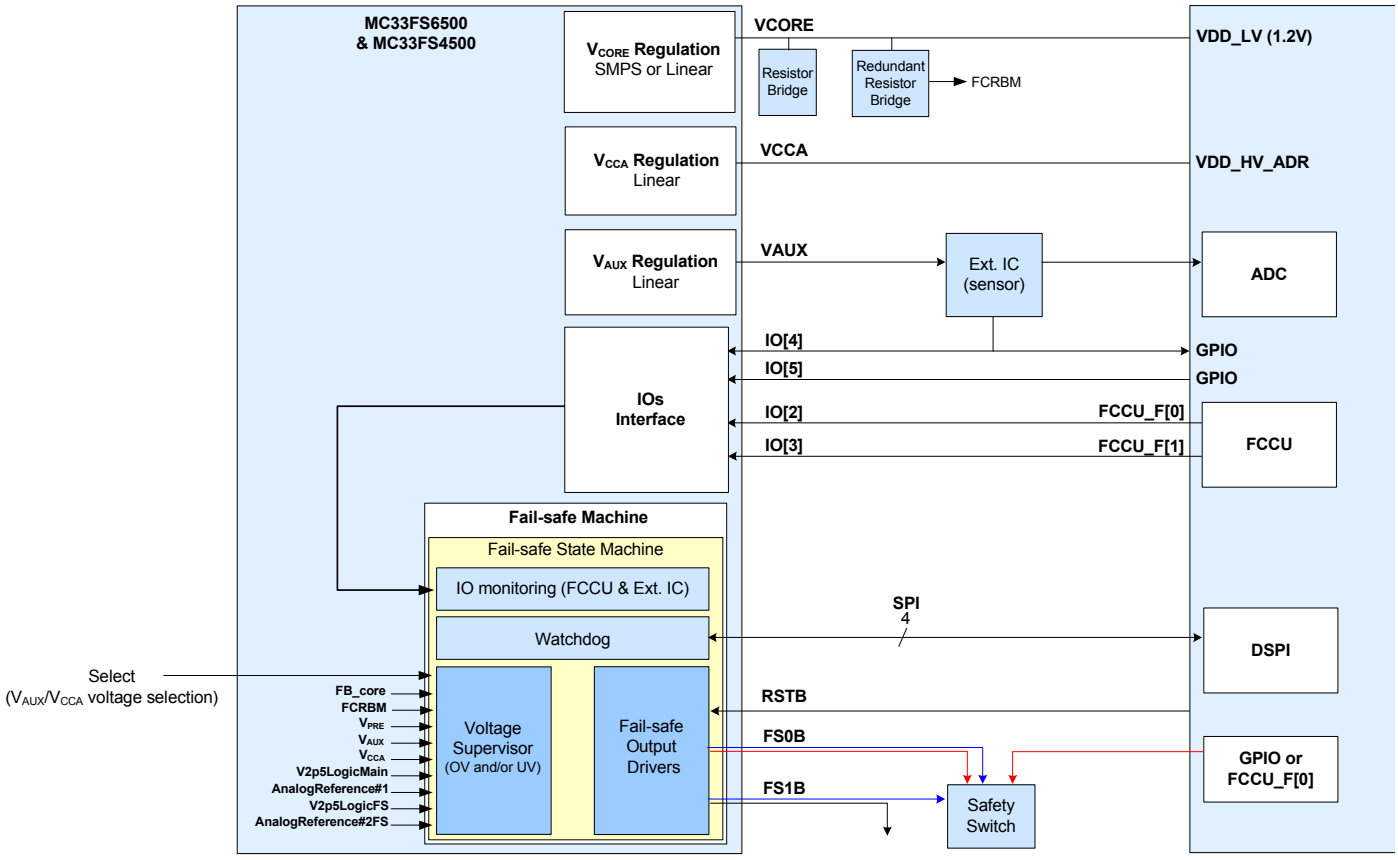


Figure 10. Functional safety-related connection to the MCU

As an example, on power steering applications, the safety switch is usually connected between the battery and the three phase MOSFET bridge driving the current to the electric motor. During a fault in the system, FS0B opens the safety switch and the electric motor is no longer powered. FS1B is activated after a delay to open the motor phase switch and avoid resistive torque going back to the steering wheel.

4.2 Functional safety integrity measures

- Replication of supply pins: three VSUP pins (VSUP1, VSUP2, and VSUP3) are used to supply the product. The loss of VSUP1 or VSUP2 shows the FS6500 and FS4500 are able to continue to work properly due to the supply redundancy.
- The Fail-safe machine is powered by VSUP3, with a redundant supply connection available through VSENSE.
- The Fail-safe machine is electrically independent from the rest of the circuit with its own oscillator, own reference voltages (Analog reference #2, V2P5_FS_Logic), and own SPI register configurations.
- The Fail-safe machine internal voltage references are monitored against overvoltage.
- Error correction or detection, or both, to reduce the effect of transient faults and permanent faults is implemented.
- Internal supplies and clock are supervised by dedicated monitors.
- Monitoring of the external voltages (FB_{CORE}, V_{CCA}, V_{AUX}) is provided through the voltage supervisor. Internal reference voltages like V2P5_Main_Analog (analog reference#1), V2P5_Main_Logic, V2P5_FS_Analog (analog reference #2), and V2P5_FS_Logic are also monitored.
- Built-in self tests (ABIST and LBIST) are implemented in hardware to detect latent faults and therefore reduce the risk of coincident latent faults (multiple-point faults).
- The FS6500 and FS4500 can react to failure notifications coming from the NXP microcontroller, using FCCU (fault collection and control unit) or external error IC monitoring.
- The risk of CMFs is reduced by a set of measures for both control and reduction of CMFs, spanning system level approaches (such as temperature and non-functional signal monitoring), physical separation, or diversity.
- The use of internal (and external) watchdogs or timeout measures.
- The safety path can be checked by asserting FS0B and FS1B independently by the SPI.
- The FCRBM pin monitors the drift of the external resistor bridge connected to FB_CORE to generate V_{CORE}.
- A dedicated mechanism is provided to measure external backup delay from external components connected to VPU_FS.

5 Safety interoperation with MCU

This section describes safety interoperation with FS6500 and FS4500 and MCU for applications requiring high functional safety integrity levels. Failure rates of external devices have to be included in the system FMEDA by the system integrator.

5.1 Device power-up

When the device is powered up, the fuses are loaded into a register bank (to fine tune internal electrical parameters) and checked by means of a hamming code. When the INIT_FS phase is closed, the Fail-safe initialization registers are also checked by means of a hamming code. If the result of this continuous check is bad due to errors which cannot be corrected, 'ERR_INT_SW' flag is set in the 'WD_ANSWER' register, and the Fail-safe outputs FS0B and FS1B are asserted low. This error detection correction measure is called SPI DED. This check is used as a safety integrity measure to detect latent faults.

Assumption: [SMR_16] It is the system integrator's responsibility to make sure the MCU checks for SPI DED errors at each WD refresh, by checking the ERR_INT_SW flag when writing in WD_ANSWER register. [END]

5.2 Power supply

5.2.1 V_{PRE}

A highly flexible SMPS pre-regulator is implemented in the FS6500 and FS4500. It can be configured as a 'non-inverting buck-boost converter' or 'standard buck converter', depending on the external configuration (low-side connection).
The pre-regulator delivers a voltage output of 6.5 V and is used internally.

5.2.1.1 Buck or buck boost configuration

An external low-side logic level MOSFET (N-type) is required to operate the 'non-inverting buck-boost converter'. The connection of the external MOSFET is detected automatically during the start-up phase.

If the low-side is not connected (GATE_LS pin connected to PGND), the product is configured as a standard buck converter.

Assumption: [SMR_17] It is the system integrator's responsibility to make sure the MCU checks the configuration of the buck/or boost configuration after system startup or after LPOFF mode, by reading LS_ DETECT bit in HW_CONFIG register.[END]

Rationale: To ensure the system can still be supplied by the device if the battery reaches a very low level during cranking (boost option).

Table 5. HW_CONFIG - LS_DETECT

Read																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	LS_DETECT	RESERVED	VCCA_PNP_DET	VCCA_HW	VAUX_HW	1	DFS_HW	DBG_HW

LS_DETECT	Description	Report the hardware configuration of V_{PRE} (buck only or buck-boost)
	0	Buck-boost
	1	Buck only
	Reset condition	Power On reset/refresh after LPOFF

5.2.1.2 V_{PRE} undervoltage

In case of a V_{PRE} undervoltage, the consequences could be visible on the other regulators available on the device. This is because the V_{PRE} is the supply of all the regulators, so by cascade effect an undervoltage can occurs on V_{CORE}, V_{CCA}, and V_{AUX}.

If the V_{PRE} undervoltage has no consequence on the other regulators, it could still have an impact on the external circuitry potentially connected to V_{PRE} (another external linear regulator for instance).

Assumption: [SMR_18] It is system integrator's responsibility to make sure the MCU checks the V_{PRE} undervoltage flag, by reading the VPRE_UV bit in the DIAG_VPRE register, in case an external circuitry is connected to the V_{PRE} as a supply of this circuitry.[END]

Rationale: To ensure the system is aware of this undervoltage and its consequences.

Table 6. DIAG_VPRE - VPRE_UV

Read																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	BoB	VPRE_STATE	TWARN_PRE	TSD_PRE	VPRE_OV	VPRE_UV	ILIM_PRE	0

VPRE_UV	Description	V_{PRE} undervoltage detection
	0	No undervoltage ($V_{PRE} > V_{PRE\_UV}$)
	1	Undervoltage detected ($V_{PRE} < V_{PRE\_UV}$)
	Reset condition	Power on reset/read

5.2.2 V_{CORE}

The FS6500 and FS4500 provide a dedicated voltage supply rail for the main input voltage of the MCU, or directly for the core of the MCU (i.e. V_{CORE}).

The voltage level of V_{CORE} supply is configurable through an external resistor bridge. The accuracy of V_{CORE} is ±2.0% without taking account the accuracy of the external resistor bridge.

It is mandatory to select an appropriate resistor tolerance for the external resistor bridge (inferior or equal to ±1.0%).

Assumption:[SMR_19] It is the system integrator's responsibility to make sure the right resistor values are well connected between V_{CORE_SNS} and ground, with the middle point connected to FB_{CORE} to configure the right voltage to the MCU.[END]

Rationale: To ensure overall operation of the MCU according to its data sheet.

Figure 11 shows how to configure an external resistor bridge for two ranges of supply level (3.3 V and 1.2 V).

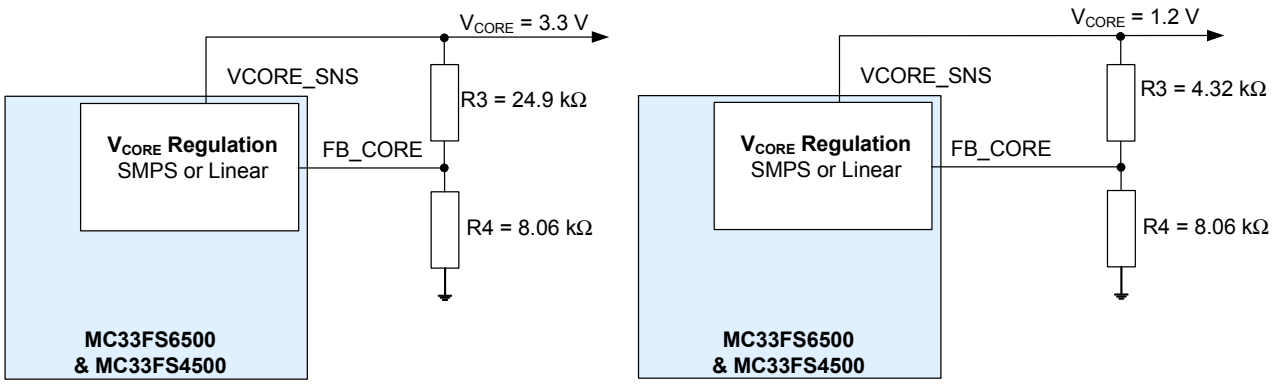


Figure 11. V_{CORE} external resistor bridge configuration

This supply voltage must be in the specified operating range of the MCU, because an overvoltage might cause permanent damage to the MCU. It is therefore either required to de-energize the MCU or to decommission/replace the MCU after an overvoltage event. An undervoltage might lead to an unexpected behavior of the MCU.

Recommendation: It is recommended at the system level to avoid V_{CORE} overvoltage and/or undervoltage, or to permanently disable (safe $state_{system}$) the system in the event of an overvoltage/undervoltage.

Rationale: To ensure overall operation of the MCU according to its data sheet.

Implementation hint: The FS6500 and FS4500 provide an overvoltage/undervoltage monitoring of the $FB_{\_CORE}$. V_{CORE} voltage is set with an external resistor bridge. If the $FB_{\_CORE}$ is above or below the value specified in the FS6500 and FS4500 data sheet, the MCU is kept powerless by switching off $FB_{\_CORE}$, and the SBC switches the system to a safe $state_{system}$ within the FTTI and maintains safe $state_{system}$ through Fail-safe outputs (FS0B, FS1B).

- Internal register configuration:
  In the FS6500 and FS4500, a register can be configured during the initialization phase to manage the impact at the system level of such overvoltage/undervoltage on V_{CORE}.
  INIT_VCORE_OVUV_IMPACT register, **VCORE_FS_OV_1:0 and VCORE_FS_UV_1:0** bits can be configured to perform actions on Fail-safe outputs if there is an overvoltage and/or undervoltage on V_{CORE}.
  By default, V_{CORE_OV} does have an impact on RSTB and FS0B, V_{CORE_UV} does have an impact on FS0B only.
  The values of overvoltage and undervoltage as well as the filtering time to avoid any sporadic detection, are specified in the FS6500 and FS4500 data sheet.
  The voltage supervisor is able to detect any spikes, oscillations, or drifts of the $FB_{\_CORE}$ voltage if the defined spikes, oscillations, or drifts are in the range of the detection capability (filtering time and voltage threshold specified on $FB_{\_CORE}$).

Table 7. INIT_VCORE_OVUV_IMPACT

Write																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	1	1	0	0	0	0	1	P	VCORE_FS_OV_1	VCORE_FS_OV_0	VCORE_FS_UV_1	VCORE_FS_UV_0	Secure_3	Secure_2	Secure_1	Secure_0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	SPI_FS_ERR	SPI_FS_CLK	SPI_FS_REQ	SPI_FS_PARITY	VCORE_FS_OV_1	VCORE_FS_OV_0	VCORE_FS_UV_1	VCORE_FS_UV_0

VCORE_FS_OV_1:0	Description	$V_{CORE\_FB\_OV}$ safety input
	00	No effect of $V_{CORE\_FB\_OV}$ on RSTB and FS0B
	01	$V_{CORE\_FB\_OV}$ DOES HAVE an impact on RSTB only
	10	$V_{CORE\_FB\_OV}$ DOES HAVE an impact on FS0B only
	11	$V_{CORE\_FB\_OV}$ DOES HAVE an impact on RSTB and FS0B
	Reset Condition	Power on reset
VCORE_FS_UV_1:0	Description	$V_{CORE\_FB\_UV}$ safety input
	00	No effect of $V_{CORE\_FB\_UV}$ on RSTB and FS0B
	01	$V_{CORE\_FB\_UV}$ DOES HAVE an impact on RSTB only
	10	$V_{CORE\_FB\_UV}$ DOES HAVE an impact on FS0B only
	11	$V_{CORE\_FB\_UV}$ DOES HAVE an impact on RSTB and FS0B
	Reset Condition	Power on reset

In addition to the V_{CORE} overvoltage/undervoltage monitoring ensured by the voltage supervisor of the FS6500 and FS4500, the device offers the possibility to detect a drift of the external resistor bridge leading to a drift of the V_{CORE} supply during system lifetime.

Assumption: [SMR_20] It is assumed the value of the external resistor bridge stays within its nominal value.[END]

Rationale: To avoid a non absolute desired voltage core supply as FB_core is checked for overvoltage and undervoltage.

Implementation hint: Connect a second external resistor bridge where the middle point is routed to the FCRBM of the FS6500 and FS4500. This second resistor bridge must be equivalent to the first resistor bridge in charge of the V_{CORE} voltage setting.

On the first resistor bridge, the FS6500 and FS4500 regulate the middle point to 0.8 V and the V_{CORE} voltage is settled according to resistor values. The second resistor bridge takes the V_{CORE} voltage value as the reference voltage and the middle point (connected to FCRBM) is measured and compared to the 0.8 V regulated by the device itself. If the difference is higher than 150 mV (maximum), the FS6500 and FS4500 bring the system in safe $state_{System}$ within the FTTI and maintain safe $state_{System}$via its safe outputs pins.

This second resistor bridge monitoring features is automatically enabled as soon as the ABIST1 execution is finished.

Figure 12 shows the good connections of the external resistor bridges.

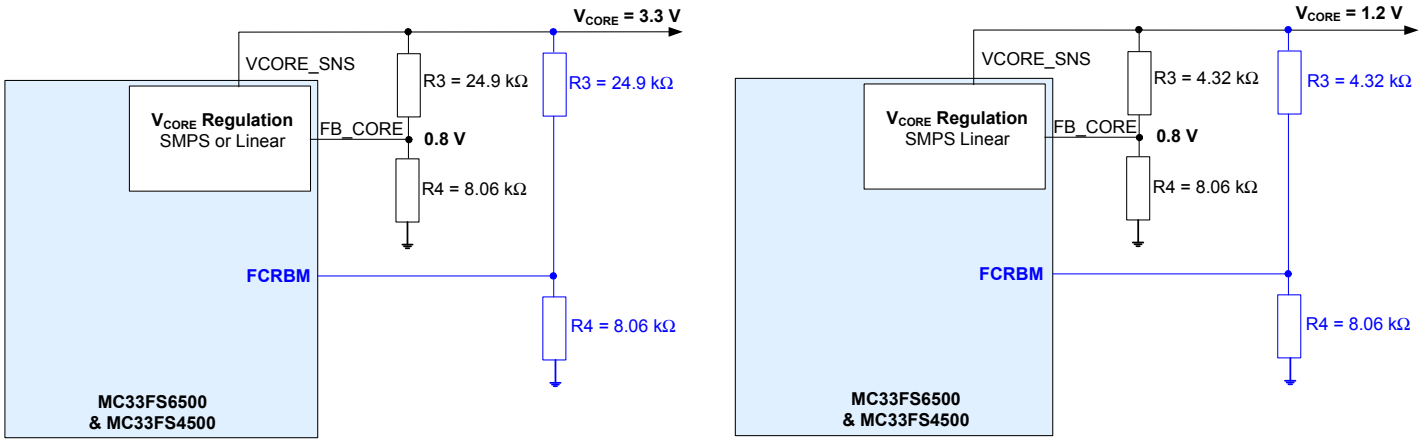


Figure 12. Connection of the second resistor bridge

If this second resistor bridge monitoring feature is not used, it is recommended to connect FCRBM pin to FB_CORE pin.

5.2.3 V_{CCA}

The FS6500 and FS4500 provide a dedicated voltage supply rail for the Analog to Digital converter of an MCU or for a local ECU supply.

The supply voltage must stay in its specified operating range, because an overvoltage might cause permanent damage to the MCU (if connected as reference voltage of an analog to digital converter, for example).

An undervoltage might lead to an unexpected behavior of the external circuitry, for example where V_{CCA} is the main supply or creates bad conversion results, if V_{CCA} is used as a reference voltage for the analog to digital converter of an MCU.

Assumption: [SMR_21] It is the system integrator's responsibility to make sure the MCU checks the VCCA output voltage level after system startup or after LPOFF mode, by reading the VCCA_HW bit in the HW_CONFIG register.[END]

Rationale: To ensure the output voltage level is as good as expected by the system requirements (3.3 V or 5.0 V).

Table 8. HW_CONFIG - VCCA_HW

Read																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	LS_DETECT	RESERVED	VCCA_PNP_DET	VCCA_HW	VAUX_HW	1	DFS_HW	DBG_HW

VCCA_HW	Description	Report the hardware configuration for VCCA
	0	3.3 V
	1	5.0 V
	Reset condition	Power on reset

Assumption: [SMR_22] It is the system integrator's responsibility to take measurements at the system level and maintain the safe $state_{system}$ when the V_{CCA} supply voltage is above or below the specified operational range.[END]

Rationale: To ensure overall operation of the analog to digital converter of the MCU or the external circuitry where V_{CCA} is connected.

Implementation hint: The FS6500 and FS4500 provide an overvoltage/undervoltage monitoring of the V_{CCA}. If V_{CCA} is above or below the value specified in the FS6500 and FS4500 data sheet, and according to its nominal value (5.0 V or 3.3 V), the MCU or external circuitry is kept powerless, and the SBC switches the system to a safe $state_{system}$ within the FTTI and maintains safe $state_{system}$ through the Fail-safe outputs (FS0B, FS1B).

- Internal register configuration:

  In the FS6500 and FS4500, a register can be configured during the initialization phase to manage the impact at the system level of such an overvoltage/undervoltage on V_{CCA}.

  INIT_VCCA_OVUV_IMPACT register, **VCCA_FS_OV_1:0 and VCCA_FS_UV_1:0** bits must be configured to perform actions on Fail-safe outputs if there is an overvoltage and/or undervoltage on V_{CCA}.

  By default, V_{CCA_OV} does have an impact on RSTB and FS0B, V_{CCA_UV} does have an impact on FS0B only.

  The values of overvoltage and undervoltage are specified in the FS6500 and FS4500 data sheet, as well as the filtering time to avoid any sporadic detection.

  The voltage supervisor is able also to detect any spikes, oscillations, or drifts on the V_{CCA} voltage, if the defined spikes, oscillations, or drifts are in the range of the detection capability (filtering time and voltage threshold specified on V_{CCA}).

Table 9. INIT_VCCA_OVUV_IMPACT

Write																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	1	1	0	0	0	0	1	P	VCCA_FS_OV_1	VCCA_FS_OV_0	VCCA_FS_UV_1	VCCA_FS_UV_0	Secure_3	Secure_2	Secure_1	Secure_0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	SPI_FS_ERR	SPI_FS_CLK	SPI_FS_REQ	SPI_FS_PARITY	VCCA_FS_OV_1	VCCA_FS_OV_0	VCCA_FS_UV_1	VCCA_FS_UV_0

VCCA_FS_OV_1:0	Description	V_{CCA_OV} safety input
	00	No effect of V_{CCA_OV} on RSTB and FS0B
	01	V_{CCA_OV} DOES HAVE an impact on RSTB only
	10	V_{CCA_OV} DOES HAVE an impact on FS0B only
	11	V_{CCA_OV} DOES HAVE an impact on RSTB and FS0B
	Reset condition	Power on reset
VCCA_FS_UV_1:0	Description	V_{CCA_UV} safety input
	00	No effect of V_{CCA_UV} on RSTB and FS0B
	01	V_{CCA_UV} DOES HAVE an impact on RSTB only
	10	V_{CCA_UV} DOES HAVE an impact on FS0B only
	11	V_{CCA_UV} DOES HAVE an impact on RSTB and FS0B
	Reset condition	Power on reset

5.2.4 V_{AUX}

The FS6500 and FS4500 provide a dedicated voltage supply rail for the IOs of an MCU. It can be configurable to supply external sensors.

This supply voltage must stay in its specified operating range, because an overvoltage might cause permanent damage to the MCU, sensors, or local ECU supply. An undervoltage might lead to an unexpected behavior of the external circuitry, for example where V_{AUX} is the main supply.

Assumption: [SMR_23] It is the system integrator's responsibility to make sure the MCU checks the VAUX output voltage level after system startup or after LPOFF mode by reading the VAUX_HW bit in the HW_CONFIG register. [END]

Rationale: To ensure the output voltage level is as good as expected by the system requirements (3.3 V or 5.0 V).

Table 10. HW_CONFIG - VAUX_HW

Read																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	LS_DETECT	RESERVED	VCCA_PNP_DET	VCCA_HW	VAUX_HW	1	DFS_HW	DBG_HW

VAUX_HW	Description	Report the hardware configuration for VAUX
	0	5.0 V
	1	3.3 V
	Reset condition	Power on reset

Assumption: [SMR_24] It is the system integrator's responsibility to take measurements at the system level and maintain the safe $state_{system}$ when the V_{AUX} supply voltage is above or below the specified operational range.[END]

Rationale: To ensure overall operation of the MCU or external circuitry (sensors) where V_{AUX} is connected.

Implementation hint: The FS6500 and FS4500 provide an overvoltage/under voltage monitoring of the V_{AUX}. If V_{AUX} is above or below the value specified in the FS6500 and FS4500 data sheet and according to its nominal value (5.0 V or 3.3 V), the MCU or external circuitry is kept powerless, and the SBC switches the system to a safe $state_{system}$ within the FTTI and maintains safe $state_{system}$ through the Fail-safe outputs (FS0B, FS1B)

- Internal register configuration:

  In the FS6500 and FS4500, a register can be configured during the initialization phase to manage the impact at the system level of such an overvoltage/undervoltage on V_{AUX}.

  INIT_VAUX_OVUV_IMPACT register, **VAUX_FS_OV_1:0 and VAUX_FS_UV_1:0** bits must be configured to perform actions on Fail-safe outputs if there is an overvoltage and/or undervoltage on V_{AUX}.

  By default, V_{AUX_OV} does have an impact on RSTB and FS0B, V_{AUX_UV} does have an impact on FS0B only.

  The values of overvoltage and undervoltage are specified in the FS6500 and FS4500 data sheet as well as the filtering time, to avoid any sporadic detection.

  The voltage supervisor is able to detect any spikes, oscillations, or drifts on the V_{AUX} voltage, if the defined spikes, oscillations, or drifts are in the range of the detection capability (filtering time and voltage threshold specified on V_{AUX})

Table 11. INIT_VAUX_OVUV_IMPACT

Write																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	1	1	0	0	0	1	0	P	VAUX_FS_OV_1	VAUX_FS_OV_0	VAUX_FS_UV_1	VAUX_FS_UV_0	Secure_3	Secure_2	Secure_1	Secure_0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	SPI_FS_ERR	SPI_FS_CLK	SPI_FS_REQ	SPI_FS_PARITY	VAUX_FS_OV_1	VAUX_FS_OV_0	VAUX_FS_UV_1	VAUX_FS_UV_0

VAUX_FS_OV_1:0	Description	V_{AUX_OV} safety input
	00	No effect of V_{AUX_OV} on RSTB and FS0B
	01	V_{AUX_OV} DOES HAVE an impact on RSTB only
	10	V_{AUX_OV} DOES HAVE an impact on FS0B only
	11	V_{AUX_OV} DOES HAVE an impact on RSTB and FS0B
	Reset condition	Power on reset

Table 11. INIT_VAUX_OVUV_IMPACT (continued)

Write		
VAUX_FS_UV_1:0	Description	V_{AUX_UV} safety input
	00	No effect of V_{AUX_UV} on RSTB and FS0B
	01	V_{AUX_UV} DOES HAVE an impact on RSTB only
	10	V_{AUX_UV} DOES HAVE an impact on FS0B only
	11	V_{AUX_UV} DOES HAVE an impact on RSTB and FS0B
	Reset condition	Power on reset

5.2.5 V_{AUX} - sensor supply

V_{AUX} can be used as a sensor supply in a system. To ensure ratiometric conversion between sensors supplied by the V_{AUX} and the analog to digital converter supplied by V_{CCA}, the V_{AUX} must be configured as a tracker of V_{CCA}. V_{AUX} can track V_{CCA} only when V_{AUX} is configured for the same output voltage as V_{CCA} (i.e for RSELECT = 5.1 kΩ or 12 kΩ. only).

Assumption: [SMR_25] It is the system integrator's responsibility to make sure the V_{CCA} linear regulator is used as reference voltage of the analog to digital converter of the MCU.[END]

Rationale: To ensure ratiometric conversion on sensors data output is powered by V_{AUX}.

Implementation hint: During the initialization phase, the **INIT_VREG** register must be configured to activate the VAUX_TRK_EN bit. V_{AUX} is then the tracker of V_{CCA} with a tracking accuracy of ±15 mV. By default the V_{AUX} tracker is not activated.

Table 12. INIT_VREG - VAUX_TRK_EN

Write																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	1	0	0	0	0	1	0	P	ICCA_LIM	TCCA_LIM_OFF	IPFF_DIS	VCAN_OV_MON	0	TAUX_LIM_OFF	VAUX_TRK_EN	0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	ICCA_LIM	TCCA_LIM_OFF	IPFF_DIS	VCAN_OV_MON	RESERVED	TAUX_LIM_OFF	VAUX_TRK_EN	BAT_FAIL
VAUX_TRK_EN	Description	Configure V_{AUX} regulator as a tracker														
	0	No tracking (default value)														
	1	Tracking enabled														
	Reset condition	Power on reset														

5.3 Safety inputs - IOs

5.3.1 IO[2] & IO[3] NXP MCU error monitoring - FCCU

The FS6500 and FS4500 monitor internal errors coming from the NXP MCU. If the MCU signals an internal failure via its error out pins (FCCU[0] and FCCU[1]), the system may no longer rely on the integrity of the device's outputs for safety functions. If an error out is indicated, the system switches and remains in the safe $state_{system}$. The SBC switches the system to a safe $state_{system}$ within the FTTI and maintains the safe $state_{system}$ through the Fail-safe outputs (FS0B, FS1B) as a reaction to the indicated error out.

Only the bi-stable protocol is covered on the FS6500 and FS4500 to use with the FCCU pins coming from the NXP microcontroller. Refer to the respective NXP MCU data sheet.

Assumption: [SMR_26] It is the system integrator's responsibility to make sure the bi-stable protocol is configured in the NXP MCU for FCCU protocol.[END]

Rationale: To monitor the NXP MCU error out signals for correct functionality of the device. The system (for example ECU) may not rely on any I/O other than FCCU_F[0] and FCCU_F[1], when those signals indicate an error.

Implementation hint: Connect the MCU FCCU_F[0] error output to the FS6500 and FS4500 IO[2] input, and the MCU FCCU_F[1] error output to the FS6500 and FS4500 IO[3] input. With the FCCU default polarity configuration, a pull-down must be connected to FCCU_F[0]/IO[2] and a pull-up must be connected to the FCCU_F[1]/IO[3], to provide a passive fault detection if the MCU does not drive its FCCU output pins. The pull-down resistor on IO[2] must be two times bigger than the pull-up resistor on IO[3] to detect IO[2] the short to IO[3] failure mode, whatever the VDDIO voltage 3.3 V or 5.0 V.

Figure 13 shows the connections of the NXP MCU FCCU and FS6500 and FS4500 IOs.

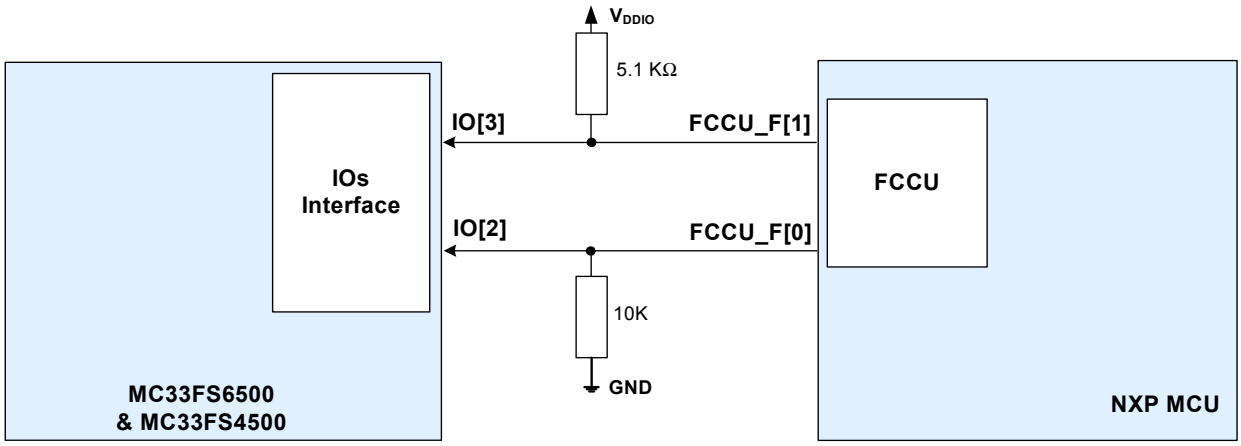


Figure 13. FCCU connection with FS6500 and FS4500 IOs

- Internal register configuration:
  In the FS6500 and FS4500, a register must be configured during initialization phase to activate the FCCU error out monitoring.
  INIT_FSSM register, **IO_23_FS** bit must be configured safety critical.
  By default, the **IO_23_FS** bit is activated.

Table 13. INIT_FSSM - IO_23_FS

Write																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	1	1	0	0	1	0	1	P	IO_45_FS	IO_23_FS	PS	RSTB_DURATION	Secure_3	Secure_2	Secure_1	Secure_0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	SPI_FS_ERR	SPI_FS_CLK	SPI_FS_REQ	SPI_FS_PARITY	IO_45_FS	IO_23_FS	PS	RSTB_DURATION

IO_23_FS	Description	Configure the couple of IO_3:2 as safety inputs for FCCU monitoring
	0	NOT SAFETY
	1	SAFETY CRITICAL
	Reset condition	Power on reset

The FCCU monitoring feature on IO_2:3 is active when the FSM is in Normal_WD mode only, If the FCCU monitoring feature on IO_2:3 is not used, the IO_23_FS bit in the INIT_FSSM register must be configured at “0” during the FSM INIT phase.

- Internal register configuration:
  In the FS6500 and FS4500, the **PS** bit in the **INIT_FSSM** register must be configured during the initialization phase to select the FCCU polarity. By default, IO_2:3 monitors active HIGH FCCU signals from the MCU.

Table 14. INIT_FSSM - PS

Write																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	1	1	0	0	1	0	1	P	IO_45_FS	IO_23_FS	PS	RSTB_DURATION	Secure_3	Secure_2	Secure_1	Secure_0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	SPI_FS_ERR	SPI_FS_CLK	SPI_FS_REQ	SPI_FS_PARITY	IO_45_FS	IO_23_FS	PS	RSTB_DURATION

PS	Description	Configure the FCCU polarity
	0	Fccu_eaout_1:0 active high
	1	Fccu_eaout_1:0 active low
	Reset condition	Power on reset

When the FCCU polarity is configured to be active low (PS bit = 1), a pull-up must be connected to FCCU_F[0]/IO[2] and a pull-down must be connected to FCCU_F[1]/IO[3] to provide a passive fault detection in case the MCU does not drive its FCCU output pins.

5.3.2 IO[2] & IO[3] non NXP MCU error monitoring

The FS6500 and FS4500 can also monitor internal errors coming from non NXP MCU with a single error out pin.

Assumption: [SMR_27] It is the system integrator's responsibility to validate non NXP MCU error monitoring implementation at the system level. [END]

Implementation hint: Connect the MCU fault module output to the FS6500 and FS4500 IO[2] input and connect IO[3] to GND with a pull-down resistor. With the FCCU default polarity configuration, a pull-down must be connected to FCCU_F[0]/IO[2] to provide a passive fault detection in case the MCU does not drive its FCCU output pins.

Figure 14 shows the connections of non NXP MCU FCCU and FS6500 and FS4500 IOs.

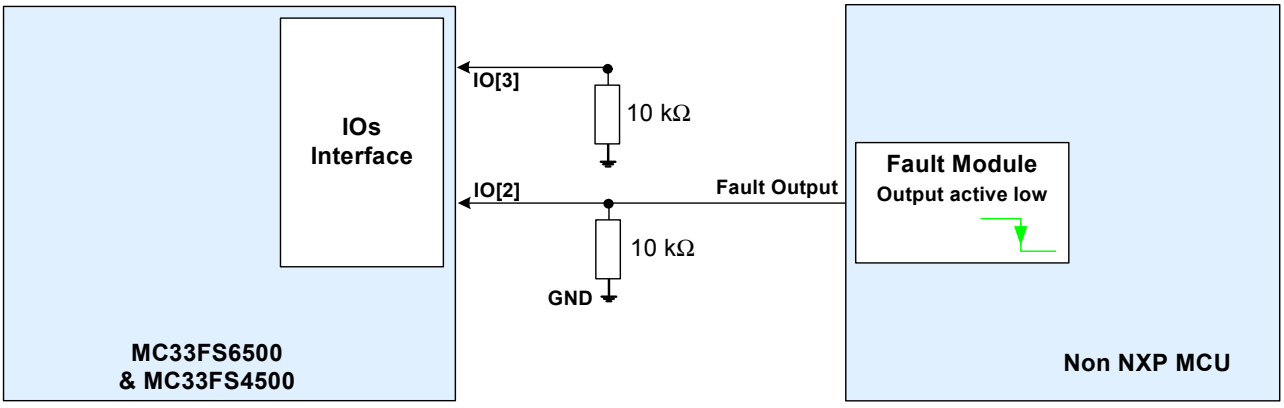


Figure 14. Single output fault module connection with FS6500 and FS4500 IOs

5.3.3 External IC error monitoring

The FS6500 and FS4500, out of the MCU FCCU monitoring, can also monitor error signals coming from an external IC. This is possible by using IO_4:5 digital inputs.

On each pair of digital inputs, one must be dedicated to monitor the output error signal coming from the external circuitry, and the other one must be connected to an output of the MCU, to listen for the acknowledgement of the error by the MCU itself.

If the external IC is not in the ECU (local) but outside (global), a serial resistor (5.1 kΩ) must be connected on the right IO to limit the input current during high transients on the line.

When an error from the external circuitry is NOT acknowledged by the MCU within a specific “acknowledgment timing”, the FS6500 and FS4500 switches the system to a safe $state_{system}$ within the FTTI and maintains the safe $state_{system}$ through the Fail-safe outputs (FS0B and FS1B).

Assumption: [SMR_28] It is the system integrator’s responsibility to make sure the error output signal from the external IC is well connected to the SBC IO[4] and one MCU GPIO, to ensure the MCU is able to listen to the fault.[END]

Rationale: Monitor a safety function realized in the ECU, out of the MCU, and bring the system to the safe $state_{system}$ during a fault.

Implementation hint: In the FS6500 and FS4500, a register can be configured during the initialization phase to manage the impact at the system level of such error monitoring using IOs by pairing out of IO[4] & IO[5].

- Internal register configuration:
  INIT FSSM register, **IO_45_FS** bits must be configured as **safety critical** to perform actions on Fail-safe outputs (FS0B and FS1B) if there is an error reported by an external IC on IO[4] and not acknowledged by the MCU on IO[5]. Refer to the FS6500 and FS4500 data sheet.
  By default, IO_45 are not configured as safety critical inputs.

Table 15. INIT_FSSM - IO_45_FS

Write																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	1	1	0	0	1	0	0	P	IO_45_FS	IO_23_FS	PS	RSTB_DURATION	Secure_3	Secure_2	Secure_1	Secure_0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	SPI_FS_ERR	SPI_FS_CLK	SPI_FS_REQ	SPI_FS_PARITY	IO_45_FS	IO_23_FS	PS	RSTB_DURATION

IO_45_FS	Description	Configure the couple of IO_5:4 as safety inputs
	0	NOT SAFETY (default value)
	1	SAFETY CRITICAL
	Reset condition	Power on reset

Figure 15 shows a connection example of an external IC error out on IO[4] connected in the ECU and the MCU acknowledgement on IO[5].

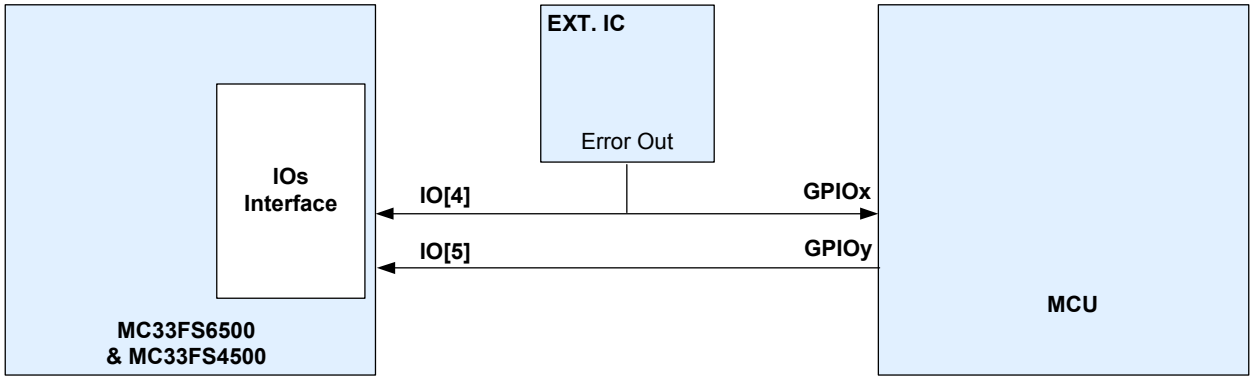


Figure 15. External IC error connection - external circuitry designed in the ECU (local)

Figure 16 shows the signal in case of an error on the external IC with or without MCU acknowledgement.

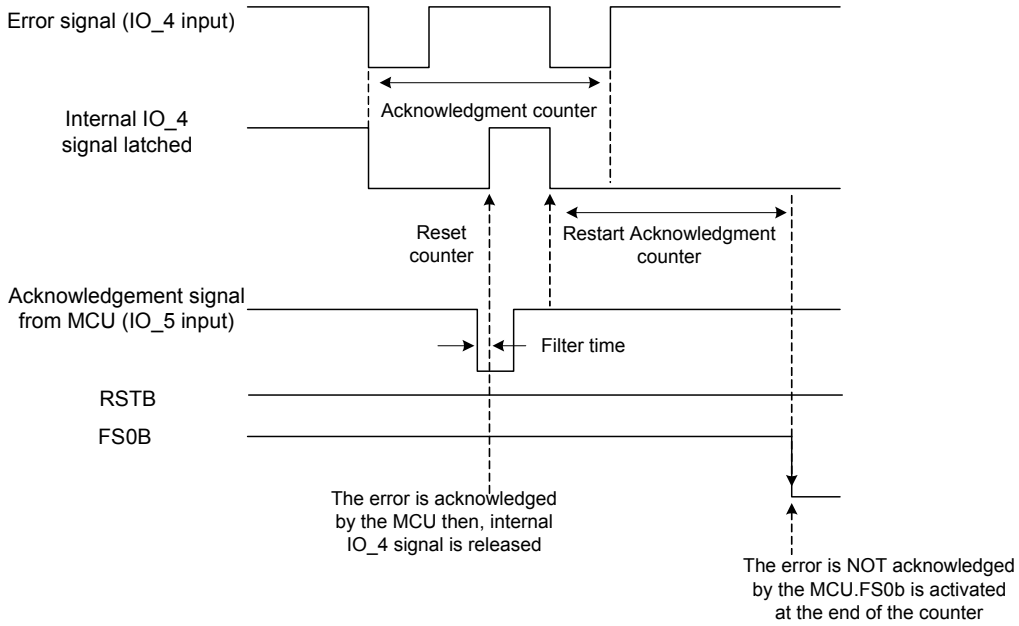


Figure 16. External IC error monitoring (timing)

Refer to the FS6500 and FS4500 data sheet for filtering time and counters.

5.3.4 How to verify the safety path in a system

When FS1B is asserted low by SPI, the t_{DELAY} configured on this pin is bypassed to avoid long diagnostic at the ECU level.

The safety path check is done when the Fail-safe machine is in normal mode since the fault counter must =0 to release the pins:

- When FS1B t_{DELAY} configuration is selected in INIT_FS phase, FS0B must be asserted before FS1B
- When FS1B t_{DUR} configuration is selected in INIT_FS phase, FS0B and FS1B assertions are independent

Figure 17 shows the connection for the safety path check at each start up.

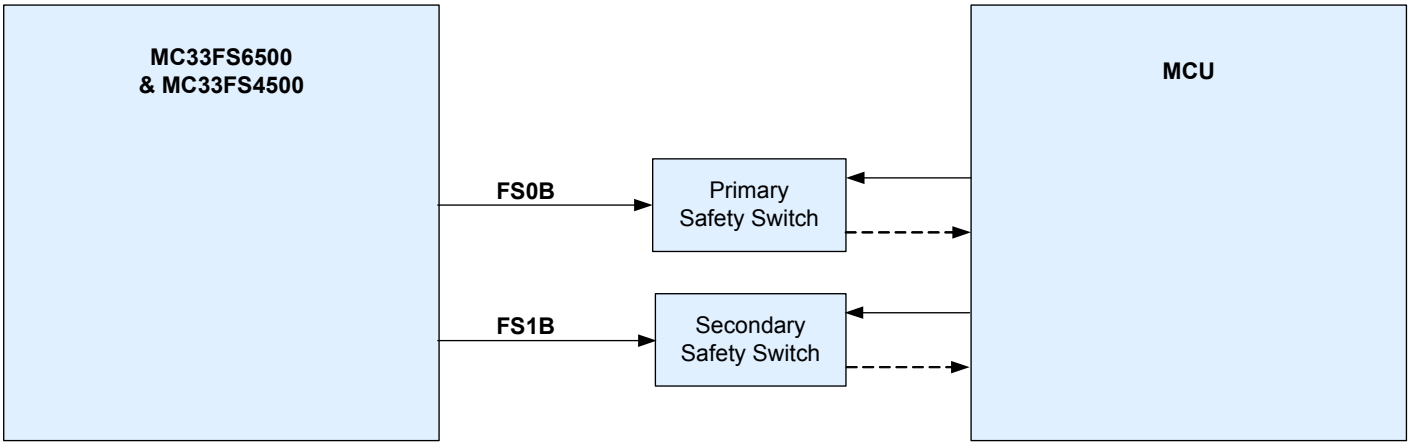


Figure 17. Safety path verification

5.3.4.1 FS0B safety path check

An FS0B activation can be requested by the SPI to check the hard connection between the FS0B pin of the SBC and the primary safety switch. This request comes from the MCU, which must monitor the good activation and release of the safety switch through a sense path from the safety switch to the MCU.

Recommendation: It is recommended to verify the safety path at each startup of the system.

Rationale: To ensure the system is well in the safe $state_{system}$ when the FS0B is asserted low before starting the application.

- Internal register:
  In the FS6500 and FS4500, a write command can be sent by the MCU to request an FS0B activation
  SF_OUTPUT_REQ register, **FS0B_REQ** bit.
  By default, the **FS0B_REQ** bit is not activated.

Table 16. SF_OUTPUT_REQ

Write																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	1	1	0	1	0	1	0	P	FS1B_REQ	FS1B_DLY_REQ	FS0B_REQ	RSTB_REQ	Secure_3	Secure_2	Secure_1	Secure_0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	SPI_FS_ERR	SPI_FS_CLK	SPI_FS_REQ	SPI_FS_PARITY	FS1B_DRV	FS1B_DLY_DRV	FS0B_DRV	RSTB_DRV

FS0B_REQ	Description	Request FS0B to be asserted low
	0	No request (default value)
	1	Request FS0B assertion
	Reset condition	Power on reset

5.3.4.2 FS1B safety path check

An FS1B activation can be requested by the SPI to check the hard connection between the FS1B pin of the SBC and the secondary safety switch. This request comes from the MCU, which must monitor the good activation and release of the safety switch through a sense path from the safety switch to the MCU.

Recommendation: It is recommended to verify the safety path at each startup of the system.

Rationale: To ensure the system is well in the safe state$_{system}$ when the FS1B is asserted low before starting the application.

- Internal register:
  In the FS6500 and FS4500, a write command can be send by the MCU to request an FS1B activation
  SF_OUTPUT_REQ register, **FS1B_REQ** bit.
  By default, the **FS1B_REQ** bit is not activated.

Table 17. SF_OUTPUT_REQ

Write																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	1	1	0	1	0	1	0	P	FS1B_REQ	FS1B_DLY_REQ	FS0B_REQ	RSTB_REQ	Secure_3	Secure_2	Secure_1	Secure_0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	SPI_FS_ERR	SPI_FS_CLK	SPI_FS_REQ	SPI_FS_PARITY	FS1B_DRV	FS1B_DLY_DRV	FS0B_DRV	RSTB_DRV

FS1B_REQ	Description	Request FS1B to be asserted low
	0	No request (default value)
	1	Request FS1B assertion (immediate assertion, no t_{DELAY})
	Reset condition	Power on reset

Figure 18 shows an example of safety path check scenario.

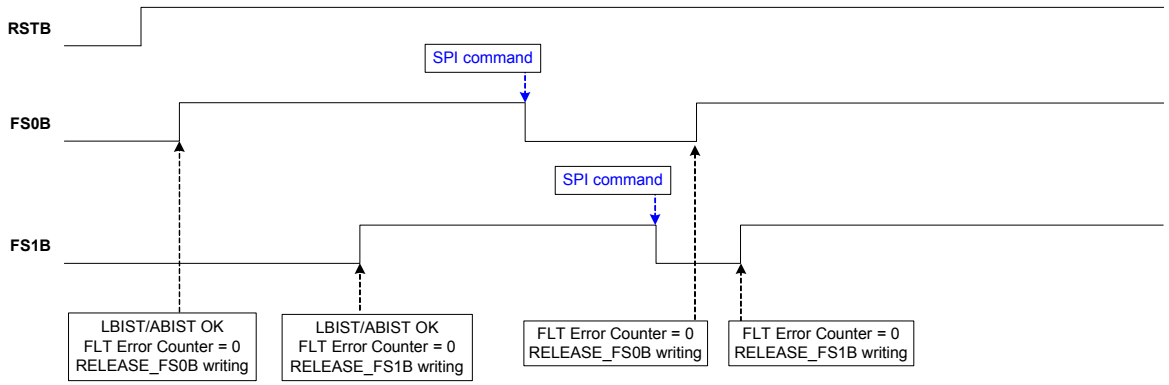


Figure 18. Safety path verification

5.4 Watchdog

A common mode may lead to a state where an MCU is unable to signal an internal failure via its FCCU error out pins (see 5.3.1, IO[2] & IO[3] NXP MCU error monitoring - FCCU, page 25). The likelihood of common mode failures affecting the functional safety of the system can be significantly reduced by using a system level timeout function (e.g. watchdog).

In general, the external watchdog covers common mode failures which are related to:

- missing/wrong power
- missing/wrong clocks
- missing/wrong resets
- general destruction of internal components (e.g. latch-up at redundant input pads)
- errors in mode change (e.g. test, debug, sleep/wake-up)

Since these errors do not result in subtle output variations of the MCU, but typically in a complete failure, a temporal watchdog is sufficient. The watchdog function is required to be sufficiently independent to the SBC (e.g regarding clock generation, power supply, implementation, etc.).

The FS6500 and FS4500 act as a supervisor of the operation, and as a consequence, include a windowed watchdog (temporal and logical monitoring) which needs to be refreshed periodically by the MCU. It means the FS6500 and FS4500 watchdog function is in permanent communication with the MCU. As soon as there is no correct communication, after repetitive and defined tries, the SBC switches the system to safe $state_{system}$ within the FTTI. Thus, either the MCU or the SBC can switch the system to safe $state_{system}$.

Assumption: [SMR_29] It is the system integrator's responsibility to make sure the MCU periodically refreshes the FS6500 and FS4500 watchdog.[END]

Rationale: To cover situations, when the MCU is not able to signal a failure.

Implementation hint: The duration of the watchdog window is configurable to allow different MCU handshake strategies. The duty cycle of the window is fixed at 50%. Therefore the first half of the window is said "closed" and the second half of the window is said 'open'. The watchdog must be refreshed in the middle of the 'open' window.

- Internal register:

  In the FS6500 and FS4500, a register can be configured during initialization phase or in normal operation. The watchdog window can be disabled during INIT_FS phase only, while the watchdog window duration can be changed in both the INIT_FS and Normal_WD phases. Doing the change in normal operation allows the system integrator to configure the watchdog window duration on the fly (the new WD window duration is taken into account when the previous one is finished).

  WD_WINDOW register, **WD_WINDOW_x** bits (where x=0 to 3) can be configured.

  By default, a window of 3.0 ms is configured which is in alignment with an FTTI below 10 ms.

Table 18. WD_WINDOW

Write																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	1	1	0	0	1	1	0	P	WD_WINDOW_3	WD_WINDOW_2	WD_WINDOW_1	WD_WINDOW_0	Secure_3	Secure_2	Secure_1	Secure_0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	SPI_FS_ERR	SPI_FS_CLK	SPI_FS_REQ	SPI_FS_PARITY	WD_WINDOW_3	WD_WINDOW_2	WD_WINDOW_1	WD_WINDOW_0

WD_WINDOW_3:0	Description	Configure the watchdog window duration. Duty cycle if set to 50%
	0000	Disable (INIT phase only)
	0001	1.0 ms
	0010	2.0 ms
	0011	3.0 ms
	0100	4.0 ms
	0101	6.0 ms
	0110	8.0 ms
	0111	12 ms
	1000	16 ms
	1001	24 ms
	1010	32 ms
	1011	64 ms
	1100	128 ms
	1101	256 ms
	1110	512 ms
	1111	1024 ms
	Reset description	Power on reset

Figure 19 shows the refresh slot allowed during WD refresh.

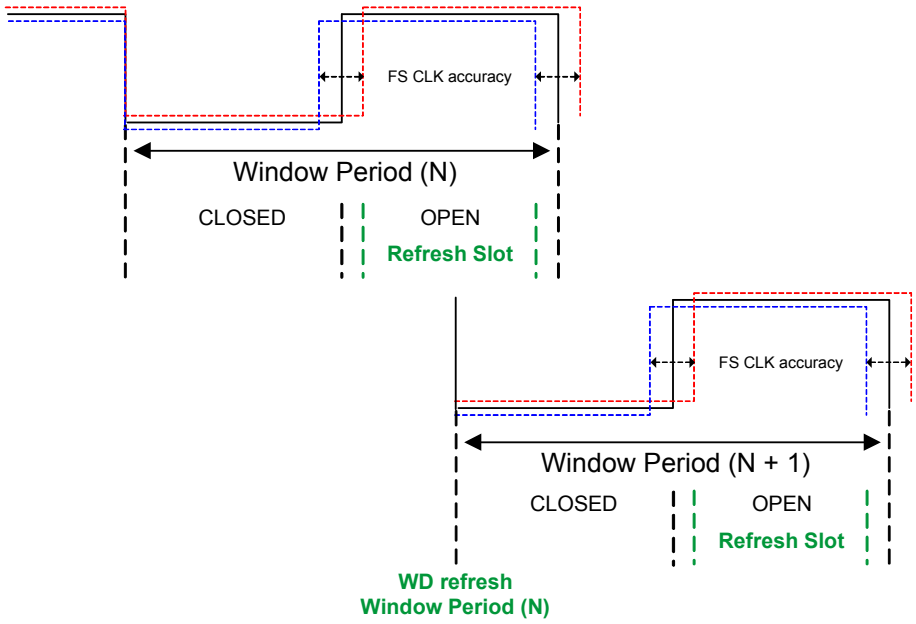


Figure 19. WD refresh slot

The windowed watchdog is based on an 8-bit pseudo-random word generated by a linear feedback shift register implemented in the SBC. A default LFSR value (0xB2) is available in the WD_LFSR register at power up or after wake-up from LPOFF. During the SBC initialization phase, the MCU can read back the LFSR to start its own calculation and then perform the watchdog answer.

The watchdog window period is derived from the Fail-safe oscillator. Refer to the FS6500 and FS4500 data sheet for the Fail-safe oscillator accuracy.

- Internal register:

  In the FS6500 and FS4500, a register can be checked during the initialization phase or even in normal operation to read back the LFSR value. It is also possible for the MCU to write its own LFSR. The new LFSR is taken by the SBC to perform its own calculation.

  WD_LFSR register, **WD_LFSR_x** bits (where x=0 to 7) - Read or write allowed.

Table 19. WD_LFSR

Read																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	WD_LFSR_7	WD_LFSR_6	WD_LFSR_5	WD_LFSR_4	WD_LFSR_3	WD_LFSR_2	WD_LFSR_1	WD_LFSR_0
WD_LFSR_7:0	Description	WD 8-bit LFSR value. Used to write the seed at any time														
	0...	bit7:bit0: 10110010 default value at start-up or after a power on reset: 0xB2														
	1...															
	Reset condition	Power on reset														

When the MCU reads the LFSR back from the FS6500 and FS4500, the MCU must start the calculation using a simple formula. Refer to the FS6500 and FS4500 data sheet. As soon as the result is available and when the window is open, the MCU must send the result to the SBC. The two results (MCU & SBC) are then compared.

- Internal register:

  In the FS6500 and FS4500, a register is available to write the result of the simple calculation based on the LFSR.

  WD_ANSWER register, **WD_ANSWER_x** bits (where x=0 to 7) - read or write allowed.

Table 20. WD_ANSWER

Write																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	1	1	0	1	0	0	1	P	WD_ANSWER_7	WD_ANSWER_6	WD_ANSWER_5	WD_ANSWER_4	WD_ANSWER_3	WD_ANSWER_2	WD_ANSWER_1	WD_ANSWER_0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	RSTB	FSxB	WD_BAD_DATA	FSO_G	IO_FS_G	WD_BAD_TIMING	ERR_INT_HW	ERR_INT_SW

WD_ANSWER_7:0	Description	WD answer from the MCU
	0...	Answer = (NOT(((LFSR x 4)+6)-4))/4
	1...	
	Reset condition	Power on reset

Table 21 shows when a watchdog answer is considered as right or wrong.

Table 21. Watchdog error

		Window	
		Closed	**Open**
SPI	**BAD key**	WD_NOK	WD_NOK
	GOOD key	WD_NOK	WD_OK
	None (timeout)	No_issue	WD_NOK

Three counters are involved each time a right or wrong watchdog refresh is performed. Refer to the FS6500 and FS4500 data sheet to understand how they interact each other.

- WD error counter
- WD refresh counter
- Fault error counter

After consecutive bad watchdog refreshes, the FS6500 and FS4500 switches the system in Fail-safe sate when the fault error counter is incremented. The watchdog impact on FS0B and RSTB pins is configurable during the INIT phase of the SBC. Correct watchdog refreshes are also monitored to allow the MCU to “get out” from a Fail-safe state when the MCU returns to an expected behavior.

The FTTI requirement at the system level (e.g. 10 ms) must be considered, and by consequence the watchdog window period must be configured accordingly. The WD error counter can also be configured to define the number of consecutive bad WD refreshes allowed by the system before tripping a reset. Basically, it can be 3, 2, or only 1.

- Internal register:

  In the FS6500 and FS4500, a register can be configured during the initialization phase to manage the impact at the system level of such a WD error.

  INIT_FS_IMPACT register, **WD_IMPACT_1:0** bits must be configured to perform actions on Fail-safe outputs if the WD error counter reaches its final value.

  By default, the WD error counter has an impact on RSTB only.

Table 22. INIT_FS_IMPACT

Write																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	1	1	0	0	1	1	0	P	TDLY_TDUR	DIS_8S	WD_IMPACT_1	WD_IMPACT_0	Secure_3	Secure_2	Secure_1	Secure_0

Table 22. INIT_FS_IMPACT

Write																
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	SPI_FS_ERR	SPI_FS_CLK	SPI_FS_REQ	SPI_FS_PARITY	TDLY_TDUR	DIS_8S	WD_IMPACT_1	WD_IMPACT_0

WD_IMPACT_1:0	Description	Watchdog impact on RSTB and/or FS0B assertion
	00	No effect on RSTB and FS0B if WD error counter = WD_CNT_ERR[1:0]
	01	RSTB only is asserted low if WD error counter = WD_CNT_ERR[1:0]
	10	FS0B only is asserted low if WD error counter = WD_CNT_ERR[1:0]
	00	RSTB and FS0B are asserted low if WD error counter = WD_CNT_ERR[1:0]
	Reset condition	Power on reset

5.5 Fault error counter

The fault error counter manages and counts the number of faults occurring in the system. The fault error counter is incremented by 1, each time the RSTB and/or FS0B pin is asserted. When FS0B is asserted, the fault error counter is incremented by 1, every time the watchdog error counter maximum value is reached.

The fault error counter has two output values (intermediate and final).

- The intermediate value can be used to force the FS0B activation or to generate a RSTB pulse, depending on the FLT_ERR_IMP_1:0 bit configuration in the INIT_FAULT register.
- The final value is used to handle the transition to deep Fail-safe if the SELECT pin is connected to GND.

Rationale: If the fault error counter reaches its final value, it means a critical permanent issue is reported at the stem level. When the deep fail-safe feature is enabled, the SBC completely switches the MCU off and maintains the system in the fail-safe state (safe $state_{system}$).

Implementation hint: In the FS6500 and FS4500, an INIT_FAULT register can be configured during the initialization phase for the intermediate and final values of the fault error counter. If the fault error counter is > 2 when FLT_ERR_FS is changed from 0 to 1, the device moves to deep fail-safe immediately. It is recommended to decrease the fault error counter value < 2 before changing the FLT_ERR_FS configuration.

- Internal register:
  INIT_FAULT register, **FLT_ERR_FS** bit and **FLT_ERR_IMP_1:0** bits
  By default, the **FLT_ERR_FS** bit is configured for an intermediate value = 3 and a final value = 6.
  By default, the **FLT_ERR_IMP_1:0** bits are configured to assert FS0B when the fault error counter ≥ intermediate value.

Table 23. INIT_FAULT - FLT_ERR_FS

Write																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	1	1	0	0	1	0	1	P	FLT_ERR_FS	FS1B_CAN_IMPACT	FLT_ERR_IMP_1	FLT_ERR_IMP_0	Secure_3	Secure_2	Secure_1	Secure_0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	SPI_FS_ERR	SPI_FS_CLK	SPI_FS_REQ	SPI_FS_PARITY	FLT_ERR_FS	FS1B_CAN_IMPACT	FLT_ERR_IMP_1	FLT_ERR_IMP_0

FLT_ERR_FS	Description	Configure the values of fault error counter
	0	Intermediate = 3; final = 6
	1	Intermediate = 1; final = 2
	Reset condition	Power on reset
FLT_ERR_IMP_1:0	Description	Configure RSTB and FS0B behavior when fault error counter ≥ intermediate value
	00	No effect on RSTB and FS0B
	01	FS0B is asserted low if FLT_ERR_CNT ≥ intermediate value
	10	RSTB is asserted low if FLT_ERR_CNT ≥ intermediate value and WD_CNT_ERR = max value
	00	FS0B is asserted low if FLT_ERR_CNT ≥ intermediate value RSTB is asserted low if FLT_ERR_CNT ≥ intermediate value and WD_CNT_ERR = max value
	Reset condition	Power on reset

Conditions which can lead to an incrementation of the fault error counter and according to the product configuration are:

- Watchdog refresh not OK or watchdog timeout during the INIT phase
- Watchdog error counter = max value (6 by default)
- VPRE overvoltage
- VCORE, VCCA, VAUX undervoltage
- VCORE, VCCA, VAUX overvoltage
- FCRBM follows VCORE configuration
- IO_23 error detection (FCCU)
- IO_45 error detection (external IC error)
- ABIST1, ABIST2 fail
- SPI DED
- External reset (except reset extension by MCU after reset assertion by the device)

The fault error counter is triplicated and a majority voter is implemented to avoid any unexpected change due to a bit flip. In case of a bit flip, the comparison is done with the two other registers and the bit in default is forced to be changed to the right value. A flag is set in the ‘ERR_INT_HW’ bit of the ‘WD_ANSWER’ register available for MCU diagnostic.

Assumption: [SMR_30] It is the system integrator’s responsibility to make sure the MCU checks the ERR_INT_HW flag at each WD refresh when writing in the WD_ANSWER register. [END]

5.5.1 Fault error counter at startup or resuming from LPOFF mode

At startup or when resuming from LPOFF mode the fault error counter starts at level 1. It is recommended to read the reset error counter and decrement it to an appropriate value by several consecutive good watchdog refreshes before a reset request by the SPI. The number of watchdogs needed (N) depends on the reset error counter value (FLT_ERR_2:0) and the WD refresh counter (WD_RFR_2:0) setup during the INIT phase. N=FLT_ERR_2:0 x (WD_RFR_2:0 + 1) to decrement the counter to “0”.

5.6 Reset output - RSTB

RSTB is a dedicated active low signal integrated into the FS6500 and FS4500 to bring the MCU under RESET during an SBC internal fault or a fault reported by the system.

In any condition, if the RSTB pin is asserted low for a duration longer than eight seconds, the device goes to:

- Deep Fail-safe if the DFS function is enabled (SELECT pin connected to Ground)
- LPOFF-sleep if the DFS function is disabled (SELECT pin connected to VPRE)

Assumption: [SMR_31] An output in high-impedance is not considered safe at the system level. It is the system integrator's responsibility to make sure external components connected to RSTB are available to bring the safety critical outputs to known levels during operation.[END]

Rationale: To bring the functional safety-critical outputs to a defined voltage level anytime.

Implementation hint: An external pull-down capacitor (filtering) and an external pull-up resistor must be connected to the right voltage rail (5.0 V or 3.3 V).

Figure 20 shows the connection of external components to insure good safety operation of RSTB.

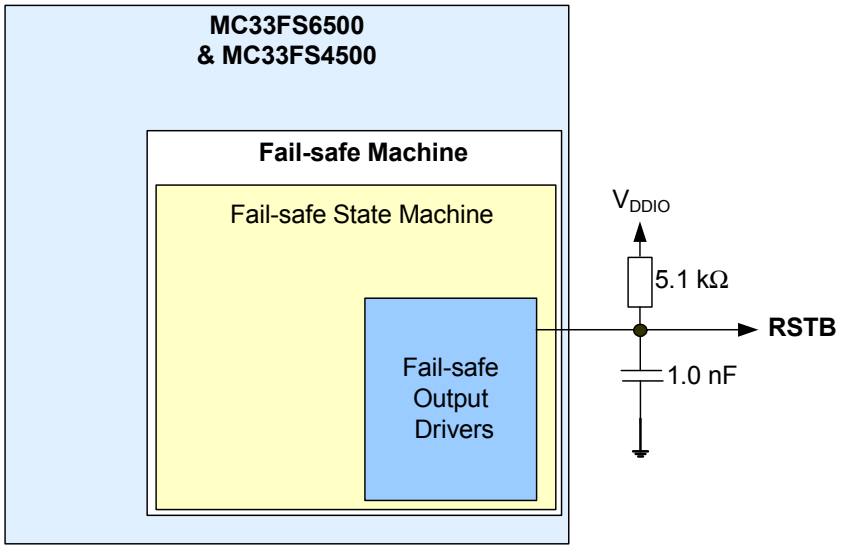


Figure 20. External components on RSTB

The duration of the reset is configurable during the initialization phase of the SBC.

- Internal register:
  In the FS6500 and FS4500, a register can be configured only during the initialization phase to define the reset duration when it is asserted low.
  INIT_FSSM register, **RSTB_DURATION** bit (1.0 ms or 10 ms low level duration available).
  By default, the reset low duration time is set to 10 ms.

Table 24. INIT_FSSM - RSTB_DURATION

Write																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	1	1	0	0	1	0	0	P	IO_45_FS	IO_23_FS	PS	RSTB_DURATION	Secure_3	Secure_2	Secure_1	Secure_0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	SPI_FS_ERR	SPI_FS_CLK	SPI_FS_REQ	SPI_FS_PARITY	IO_45_FS	IO_23_FS	PS	RSTB_DURATION

RSTB_DURATION	Description	Configure the reset low duration time
	0	10 ms
	1	1.0 ms
	Reset condition	Power on reset

RSTB low pulse can also be requested by the SPI to check for a hard connection between the MCU reset pin and the FS6500 and FS4500. This request comes from the MCU itself and is a software request. This action must be done before releasing the FS0B to a high. The goal is to verify the good RSTB hardware connection between the MCU and the FS6500 and FS4500.

- Internal register:
  In the FS6500 and FS4500, a write command can be sent by the MCU to request a low reset pulse.
  SF_OUTPUT_REQ register, **RSTB_REQ** bit.
  By default, the **RSTB_REQ** bit is not activated.

Table 25. SF_OUTPUT_REQ

Write																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	1	1	0	1	0	1	0	P	FS1B_REQ	FS1B_DLY_REQ	FS0B_REQ	RSTB_REQ	Secure_3	Secure_2	Secure_1	Secure_0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	SPI_FS_ERR	SPI_FS_CLK	SPI_FS_REQ	SPI_FS_PARITY	FS1B_DRV	FS1B_DLY_DRV	FS0B_DRV	RSTB_DRV

RSTB_REQ	Description	Request a RSTB low pulse
	0	No request (default value)
	1	Request a RSTB low pulse
	Reset condition	Power on reset / When RSTB is done

The RSTB pin is bi-directional, so the FS6500 and FS4500 can bring the MCU under RESET and the MCU can maintain the RSTB low even if the FS6500 and FS4500 is ready to release it. All the reset numbers asserted by the FS6500 and FS4500 are populated in the fault error counter.

5.6.1 RSTB internal monitoring

An internal sense path of the RSTB pin is implemented in the device. The function monitors the output of the pin and compares it with the digital command. If a difference between the digital command and the RSTB internal sense path is detected, the failure to bring the system to a safe state by the RSTB pin is reported (FS0_G bit in the WD_ANSWER register, and RSTB_DIAG bit in the DIAG_SF_IOs register) and the FS0B pin is asserted low (i.e. external short to high when the device asserts the RSTB low).

5.7 Safety outputs - FS0B, FS1B

The safety outputs are used to switch the system to the Fail-safe state (safe $state_{system}$). An integrated pull-down resistor guarantee a passive low level to maintain the Fail-safe sate (safe $state_{SBC}$) even when the device is completely unpowered.

5.7.1 FS0B

FS0B is a dedicated active low signal integrated in the FS6500 and FS4500 to bring the system to the Fail-safe state (safe $state_{system}$) when needed. This safety output can be used for opening the power supply line, disabling the half-bridge drive of a Motor/Valve,...

Assumption: [SMR_32] An output in high-impedance is not considered safe at the system level. It is the system integrator's responsibility to make sure external components connected to FS0B are available to bring the safety critical outputs to a known level during operation. [END]

Rationale: To bring the functional safety-critical outputs to a defined voltage level at anytime.

Implementation hint: An external pull-down capacitor (filtering) and an external pull-up resistor must be connected to the right voltage rail (up to battery voltage).

Assumption: [SMR_33] It is the system integrator's responsibility to ensure opening the safety switch in a system must not be driven by the FS0B only, but also by the MCU or FS1B.[END]

Rationale: To have a redundant path to cover an external FS0B short to high failure mode.

Implementation hint: A redundant signal coming from the MCU (in red) with a pull-down resistor to cover passive sate when the MCU is in reset or FS1B coming from FS6500 and FS4500 (in blue).

Figure 21 shows the connection of external components to ensure good safety operation of FS0B.

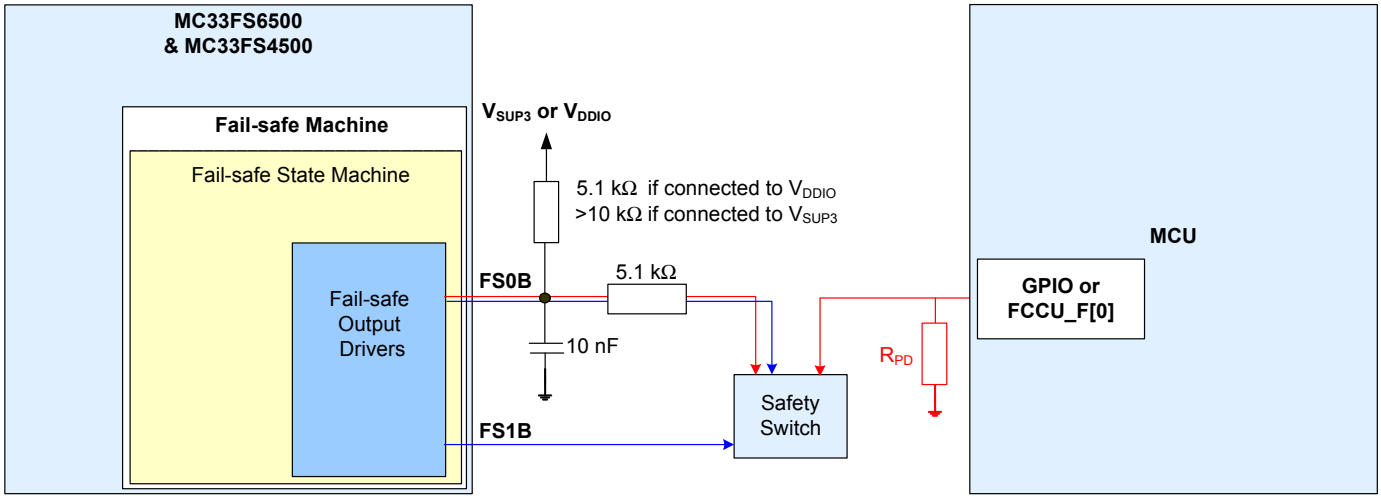


Figure 21. External components on FS0B and redundant safety path from the MCU or FS1B

Assumption: [SMR_34] It is system integrator's responsibility to make sure a resistor in series is well connected to FS0B.[END]

Rationale: FS0B can be connected externally to the system. In this case, it must be robust against automotive transients which can appear on the battery line.

Implementation hint: A resistor of 5.1 kΩ must be connected in series on FS0B.

5.7.2 FS1B

FS1B is a dedicated active low signal integrated into the FS6500 and FS4500 which can be activated after a delay (t_{DELAY}) or for a duration (t_{DUR}) when FS0B is activated. FS1B can also be used as an FS0B redundant safety output when t_{DELAY} = 0. In this case, both FS0B and FS1B pins are asserted low at the same time.

This safety output can be used to open the phases of a motor after demagnetization of the coils, to disable an external physical layer for a certain duration to avoid miscommunication when a failure happens, or any other use case where a second safety output is required.

Assumption: [SMR_35] An output in high-impedance is not considered safe at the system level. It is the system integrator's responsibility to make sure external components connected to FS1B are available to bring the safety critical outputs to a known level during operation. [END]

Rationale: To bring the functional safety-critical outputs to a defined voltage level at anytime.

Implementation hint: An external pull-down capacitor (filtering) and an external pull-up resistor must be connected to the right voltage rail (VPU_FS). VPU_FS offers an independent pull-up on FS1B compared to FS0B, and avoids common cause failures to guarantee the delay between FS0B and FS1B activations in all conditions.

Figure 22 shows the connection of external components to ensure good safety operation of FS1B.

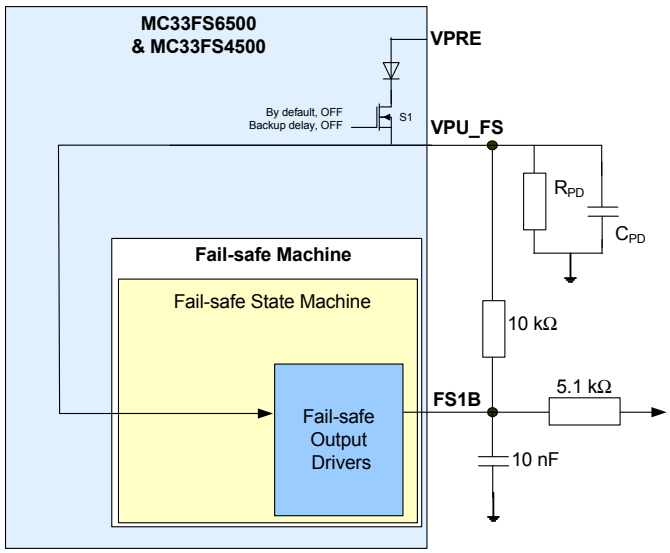


Figure 22. External components on FS1B

Assumption: [SMR_36] It is system integrator's responsibility to make sure a resistor in series is well connected to FS1B.[END]

Rationale: FS1B can be connected externally to the system. In this case, it must be robust against automotive transients which can appear on the battery line.

Implementation hint: A resistor of 5.1 kΩ must be connected in series on FS1B.

FS1B mode, t_{DELAY} or t_{DUR}, is configurable during initialization phase of the SBC.

- Internal register:
  INIT_FS_IMPACT register, **TDLY_TDUR** bit.
  By default, the **TDLY_TDUR** bit configures FS1B in t_{DELAY} mode.

Table 26. INIT_FS_IMPACT

Write																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	1	1	0	0	1	1	0	P	TDLY_TDUR	DIS_8S	WD_IMPACT_1	WD_IMPACT_0	Secure_3	Secure_2	Secure_1	Secure_0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	SPI_FS_ERR	SPI_FS_CLK	SPI_FS_REQ	SPI_FS_PARITY	TDLY_TDUR	DIS_8S	WD_IMPACT_1	WD_IMPACT_0

TDLY_TDUR	Description	FS1B delay or FS1B duration mode selection
	0	FS1B delay mode
	1	FS1B duration mode
	Reset condition	Power on reset

FS1B delay and duration timings are configurable via the SPI during initialization phase of the SBC from 0 ms to 3150 ms with the combination of FS1B_TIME_3:0 and FS1B_TIME_RANGE bits. These timings are derived from the Fail-safe oscillator. Refer to the FS6500 and FS4500 data sheet for the Fail-safe oscillator accuracy.

- Internal register:
  INIT_FS1B_TIMING register, **FS1B_TIME_x** bits (where x=0 to 3).
  By default, a timing of 37 ms is configured (FS1B_TIME_RANGE = 0).

Table 27. INIT_FS1B_TIMING

Write																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	1	1	0	0	0	0	1	P	FS1B_TIME_3	FS1B_TIME_2	FS1B_TIME_1	FS1B_TIME_0	Secure_3	Secure_2	Secure_1	Secure_0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	SPI_FS_ERR	SPI_FS_CLK	SPI_FS_REQ	SPI_FS_PARITY	FS1B_TIME_3	FS1B_TIME_2	FS1B_TIME_1	FS1B_TIME_0

Table 27. INIT_FS1B_TIMING (continued)

Write			
FS1B_TIME_3:0	Description	FS1B Timing Range Factor x1 (FS1B_TIME_RANGE bit = 0)	FS1B Timing Range Factor x8 (FS1B_TIME_RANGE bit = 1)
	0000	0	0
	0001	10 ms	80 ms
	0010	13 ms	104 ms
	0011	17 ms	135 ms
	0100	22 ms	176 ms
	0101	29 ms	228 ms
	0110	37 ms	297 ms
	0110	48 ms	386 ms
	1001	63 ms	502 ms
	1010	82 ms	653 ms
	1011	106 ms	848 ms
	1100	138 ms	1103 ms
	1101	179 ms	1434 ms
	1110	233 ms	1864 ms
	1110	303 ms	2423 ms
	1001	394 ms	3150 ms
	Reset condition	Power on reset	

- Internal register:
  INIT_SUPERVISOR register, **FS1B_TIME_RANGE** bit.
  By default, a x1 timing range factor is applied to FS1B_TIME timing.

Table 28. INIT_SUPERVISOR

Write																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	1	1	0	0	0	1	1	P	VCORE_5D	VCCA_5D	VAUX_5D	FS1B_TIME_RANGE	Secure_3	Secure_2	Secure_1	Secure_0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	SPI_FS_ERR	SPI_FS_CLK	SPI_FS_REQ	SPI_FS_PARITY	VCORE_5D	VCCA_5D	VAUX_5D	FS1B_TIME_RANGE

FS1B_TIME_RANGE	Description	Configure the FS1B timing range factor x1 or x8
	0	x1 timing range factor
	1	x8 timing range factor
	Reset condition	Power on reset

5.7.2.1 Backup delay with external components

When FS1B is configured to be asserted with a delay after FS0B assertion, the delay is generated by the Fail-safe oscillator based on the FS1B_TIME timing selected during initialization phase, except in cases of loss of power supply or loss of the Fail-safe oscillator. If one of these two faults is detected, the switch S1 is opened and a backup delay is generated by R_{PD}/C_{PD} external components connected to the VPU_FS pin. R_{PD}/C_{PD} external components are required only for the backup delay implementation.

Assumption: [SMR_37] It is system integrator's responsibility to make sure R_{PD} and C_{PD} are well connected to VPU_FS.[END]

Rationale: To prevent latent faults from R_{PD}/C_{PD} external components

Implementation hint: The MCU sets the SPI bit FS1B_DLY_REQ in the SF_OUTPUT_REQ register. This bit opens the S1 switch and engages the backup delay. FS1B is supplied by R_{PD}/C_{PD} external components from VPU_FS and the discharge timing is controlled by FS1B assertion when FS1B = 3.0 V (typical) (see Figure 23). After the expected backup delay, the MCU reads the bit FS1B_SNS in the RELEASE_FSxB register to verify FS1B assertion.

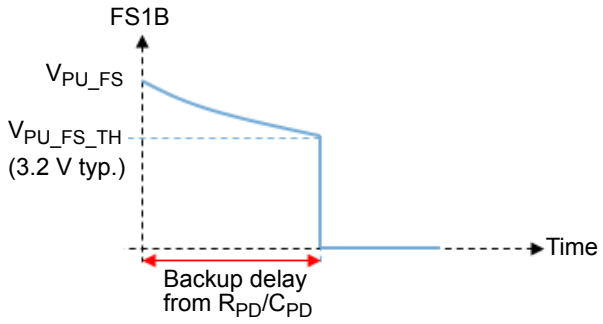


Figure 23. FS1B activation by the backup delay

Refer to the FS6500 and FS4500 data sheet (chapter t_{DELAY} operation) for more details on how to calculate R_{PD}/C_{PD} values for a desired backup delay.

Recommendation: At each startup of the system or before going to LPOFF, it is recommended to exercise the backup delay.

Note: After FS1B assertion by the digital FS1B_TIME delay, if the analog backup delay is activated due to a loss of power supply or loss of Fail-safe oscillator (considered as a multiple point fault), a glitch is generated at FS1B from V_{PU_FS} to the $V_{PU\_FS\_TH}$ voltage for the backup delay duration.

- Internal register:
  In the FS6500 and FS4500, a write command can be sent by the MCU to request an FS1B backup delay activation.
  SF_OUTPUT_REQ register, **FS1B_DLY_REQ_REQ** bit.
  By default, the **FS1B_DLY_REQ** bit is not activated.

Table 29. SF_OUTPUT_REQ

Write																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	1	1	0	1	0	1	0	P	FS1B_REQ	FS1B_DLY_REQ	FS0B_REQ	RSTB_REQ	Secure_3	Secure_2	Secure_1	Secure_0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	SPI_FS_ERR	SPI_FS_CLK	SPI_FS_REQ	SPI_FS_PARITY	FS1B_DRV	FS1B_DLY_DRV	FS0B_DRV	RSTB_DRV

FS1B_DLY_REQ	Description	Request activation of FS1B backup delay
	0	No request (default value)
	1	Request FS1B assertion (with t_{DELAY} controlled by the backup delay)
	Reset condition	Power on reset

- Internal register:
  In the FS6500 and FS4500, a read command can be sent by the MCU to verify the FS1B die pad state.
  RELEASE_FSxB register, **FS1B_SNS** bit.

Table 30. RELEASE_FSxB

Read																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	SPI_FS_ERR	SPI_FS_CLK	SPI_FS_REQ	SPI_FS_PARITY	RESERVED	FS1B_SNS	FS0B_SNS	RSTB_SNS

FS1B_SNS	Description	Sense of FS1B pad
	0	FS1B pad sense low
	1	FS1B pad sense high
	Reset condition	Power on reset

5.7.3 FS0B, FS1B internal monitoring

An internal sense path of the FS0B and FS1B pins is implemented in the device. The function monitors the output of each pin and compares it with the digital command.

- If a difference between the digital command and the FS0B internal sense path is detected, the failure to bring the system to a safe state by FS0B is reported (FS0_G bit in the WD answer register) and the RSTB pin is asserted low. (i.e. external short to high when the device asserts the FS0B low).
- If a difference between the digital command and the FS1B internal sense path is detected, the failure to bring the system to a safe state by FS1B is reported (FS0_G bit in the WD answer register) and the FS0B pin is asserted low. (i.e. external short to high when the device asserts the FS1B low).

5.7.4 Release of FS0B & FS1B

When the Fail-safe outputs FS0B and consequently FS1B are asserted low by the device due to a fault, some conditions must be validated before allowing these pins to be released by the device. The conditions to leave the safe state$_{SBC}$ are:

- the fault is removed.
- the fault error counter must be at 0.
- the S1 switch to connect the VPU_FS pull-up to V_{PRE} must be closed before releasing FS1B.
- the RELEASE_FSxB register must be filled with the right value.

Refer to the FS6500 and FS4500 Data Sheet (chapter RELEASE_FSxB Register) to know how to fill in the RELEASE_FSxB register and release FS0B and FS1B independently or simultaneously.

5.8 Built-in hardware self tests (BIST)

Built-in hardware self-test (BIST) is a mechanism permitting circuitry to test itself. Not every fault expresses itself immediately. For example, a fault may remain unnoticed if a component is not used, the context is not causing an error, or the error is masked.

If faults are not detected over a long time (latent faults), they can pile up once they propagate. ISO 26262 requires a latent-fault metric for ASIL D ≥ 90%. Typically hardware assisted BIST is therefore used as a safety integrity measure to detect latent faults.

The FS6500 and FS4500 is equipped with a built-in hardware self-test:

- Logic BIST (LBIST), executed at startup and going out from LPOFF mode
  - during LBIST the device tests the functional logic of the Fail-safe machine against a stuck at fault (coverage >90%)
- Analog BIST1 (ABIST1), executed at start-up and going out from LPOFF mode
  - during ABIST1 the product tests the analog monitoring functions showed in Table 31
- Analog BIST2 (ABIST2), executed on demand by the SPI request from the MCU
  - during ABIST2 the product tests the analog monitoring functions showed in Table 33

5.8.1 LBIST and ABIST1

A logic BIST and an analog BIST are performed automatically after start-up and resuming from LPOFF mode to ensure the integrity of the system. If a latent fault is detected, the RSTB pin is released for diagnostic, but the FS0B and FS1B pins remain asserted low to maintain the application in safe state.

Table 31. ABIST1 checks

Parameters	Overvoltage	Undervoltage	Short to high	Low speed	High speed	Comments
V_{PRE}	X					
V_{CORE}	X	X				
V_{CCA}	X	X				
V2P5 Main digital	X					Undervoltage not checked because undervoltage means power on reset state
V2P5 Main analog	X					Undervoltage not checked because undervoltage means power on reset state
V2P5 Fail-safe digital	X					Undervoltage not checked because undervoltage means power on reset state
V2P5 Fail-safe analog	X					Undervoltage not checked because undervoltage means power on reset state
Osc Fail-safe				X	X	
FCRBM	X	X				
RSTB			X			Internal sense path is checked for high and low level
FS0B			X			Internal sense path is checked for high and low level

Assumption: [SMR_38] It is the system integrator's responsibility to make sure the MCU checks to see both LBIST and ABIST1 PASS after start-up and resuming from LPOFF mode.[END]

Rationale: To prevent latent faults.

Implementation hint: Read LBIST_OK and ABIST1_OK bits in the BIST register.

Table 32. BIST - LBIST and ABIST1 diagnostic results

Read																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	SPI_FS_ERR	SPI_FS_CLK	SPI_FS_REQ	SPI_FS_PARITY	LBIST_OK	ABIST2_FS1B_OK	ABIST2_VAUX_OK	ABIST1_OK

LBIST_OK	Description	Diagnostic of Fail-safe logic BIST (automatically executed)
	0	LBIST fail
	1	LBIST pass
	Reset condition	Fail-safe power on reset
ABIST1_OK	Description	Diagnostic of analog BIST1 (automatically executed)
	0	ABIST1 fail
	1	ABIST1 pass
	Reset condition	Fail-safe power on reset

5.8.2 ABIST2 on FS1B and VAUX

A second Analog BIST on FS1B and VAUX can be performed on demand after start-up and resuming from LPOFF mode to ensure the integrity of the system. ABIST2 must be executed and pass for FS1B, and V_{AUX} when V_{AUX} is declared safety critical (overvoltage and/or undervoltage have an impact on Fail-safe outputs) to release the FS0B pin.

ABIST2 on FS1B and VAUX can be launched separately or simultaneously. If a latent fault is detected, the FS0B and FS1B pins remain asserted low to maintain the application in safe state.

Table 33. ABIST2 checks

Parameters	Overvoltage	Undervoltage	Short to high	Low speed	High speed	Comments
V_{AUX}	X	X				
FS1B			X			Internal sense path is checked for high and low level

Assumption: [SMR_39] It is the system integrator's responsibility to make sure the MCU executes the ABIST2 by the SPI request and checks both FS1B and VAUX ABIST are a pass by reading the ABIST2_FS1B_OK and ABIST2_VAUX_OK bits in the BIST register.[END]

Rationale: To prevent latent faults.

Implementation hint: Write ABIST2_FS1B and ABIST2_VAUX bits in the BIST register. Wait 200 µs. Read ABIST2_FS1B_OK and ABIST2_VAUX_OK in the BIST register.

Recommendation: At start-up and after each wake-up event from LPOFF mode, it is recommended to execute ABIST2 on both FS1B and VAUX by the SPI request from the MCU during INIT phase.

- Internal register:
  BIST register, **ABIST_FS1B and ABIST_VAUX** bits.
  By default, ABIST2 on FS1B and VAUX is not executed.

Table 34. BIST - ABIST2 request

Write																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	1	1	0	0	0	1	0	P	0	ABIST2_FS1B	ABIST2_VAUX	0	Secure_3	Secure_2	Secure_1	Secure_0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	SPI_FS_ERR	SPI_FS_CLK	SPI_FS_REQ	SPI_FS_PARITY	LBIST_OK	ABIST2_FS1B_OK	ABIST2_VAUX_OK	ABIST1_OK

ABIST2_FS1B	Description	Request ABIST execution on FS1B
	0	No action (default value)
	1	Launch ABIST on FS1B
	Reset condition	Fail-safe power on reset
ABIST2_VAUX	Description	Request ABIST execution on VAUX
	0	No action (default value)
	1	Launch ABIST on VAUX
	Reset condition	Fail-safe power on reset

Table 35. BIST - ABIST2 diagnostic results

Read																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	SPI_FS_ERR	SPI_FS_CLK	SPI_FS_REQ	SPI_FS_PARITY	LBIST_OK	ABIST2_FS1B_OK	ABIST2_VAUX_OK	ABIST1_OK

ABIST2_FS1B_OK	Description	Diagnostic of FS1B Analog BIST2 (executed on demand)
	0	FS1B ABIST FAIL or NOT executed
	1	FS1B ABIST PASS
	Reset condition	Fail-safe power on reset
ABIST2_VAUX_OK	Description	Diagnostic of VAUX Analog BIST2 (executed on demand)
	0	VAUX ABIST FAIL or NOT executed
	1	VAUX ABIST PASS
	Reset condition	Fail-safe power on reset

5.9 Deep fail-safe

Deep fail-safe function is enabled when the SELECT pin is connected to ground and disabled when the SELECT pin is connected to VPRE. The configuration is done after each power on reset, and after a wake-up event when device is in LPOFF by both the Main and the Fail-safe logics.

When the deep fail-safe function is enabled, as soon as either the fault error counter reaches its final value or the RSTB pin remains asserted low for more than 8.0 s, the device switches all the regulators off (except VKAM if VKAM was on) and maintain the safe state$_{SBC}$ (FS0B and FS1B are activated). Only a Key_OFF/Key_ON sequence (IO_0=0 followed by IO_0=1) at the system level can recover the situation.

Assumption: [SMR_40] It is the system integrator's responsibility to make sure the MCU checks to see the deep fail-safe feature is activated after system startup or after LPOFF mode, by reading both the DFS_HW1 bit in HW_CONFIG register (Main logic detection) and DFS_HW2 bit in DEVICE_ID_FS register (Fail-safe logic detection).[END]

Rationale: To ensure the product moves to deep Fail-safe as expected when the SELECT pin is connected to GND.

Table 36. HW_CONFIG - DFS_HW1

Read																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	LS_DETECT	RESERVED	VCCA_PNP_DET	VCCA_HW	VAUX_HW	1	DFS_HW1	DBG_HW

DFS_HW1	Description	Report the configuration of the deep Fail-safe feature on the Main logic
	0	Deep Fail-safe enable
	1	Deep Fail-safe disable
	Reset condition	Power on reset / refresh after LPOFF

Table 37. DEVICE_ID_FS - DFS_HW2

Read																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	DFS_HW2	FS1

DFS_HW2	Description	Report the configuration of the deep Fail-safe feature on Fail-safe logic
	0	Deep fail-safe enable
	1	Deep fail-safe disable
	Reset condition	Power on reset / refresh after LPOFF

5.10 Debug mode

A debug mode is available on the device to help the system engineer to develop its software. Refer to the FS6500 and FS4500 data sheet (chapter DEBUG input (entering in debug mode)) to learn how to enter and exit debug mode.

In debug mode, the watchdog window is fully open and no watchdog refresh is required. This allows an easy debug of the hardware and software routines (i.e SPI commands). However, the whole watchdog functionality is kept on (seed, LFSR, WD refresh counter, WD error counter,...). WD errors are detected and counted with reaction according to WD_IMPACT bit configuration.

When the debug mode is activated, the CAN transceiver is set to normal operation mode. This allows communication with the MCU, in case the SPI communication is not available (the case of MCU not programmed).

Assumption: [SMR_41] It is the system integrator's responsibility to make sure the MCU checks to see the mode is not activated after system startup or after LPOFF mode by reading the DBG_HW bit in HW_CONFIG register.[END]

Rationale: To ensure the product is not working in debug mode and by consequence is able to assert the safety outputs RSTB and/or FS0B low, in case of a watchdog refresh error.

Table 38. HW_CONFIG - DBG_HW

Read																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	LS_DETECT	RESERVED	VCCA_PNP_DET	VCCA_HW	VAUX_HW	1	DFS_HW	DBG_HW

DBG_HW	Description	Report the configuration of the debug mode
	0	Normal operation
	1	Debug mode selected
	Reset Condition	Power on reset / refresh after LPOFF

To activate the normal mode at startup, the debug pin of the device must have an external pull-down to GND. Refer to the FS6500 and FS4500 data sheet.

5.11 Physical layers

LIN and FS1B functions are exclusive. The differentiation is made by part numbers. When LIN is available, FS1B is not, and vice versa.

5.11.1 LIN mode during RSTB assertion

When RSTB is asserted low, the LIN physical layer is automatically disabled to avoid miscommunication (LIN_MODE_1:0 = 00, Sleep/no wake-up capability). When RSTB is released, the LIN physical layer is automatically enabled and its configuration depends on the LIN_MODE_1:0 bits.

5.11.2 CAN mode during RSTB and FS1B assertion

When RSTB is asserted low, the CAN physical layer is automatically disabled to avoid miscommunication (CAN_MODE_1:0 = 00, Sleep/no wake-up capability). When RSTB is released, the CAN physical layer is automatically enabled and its configuration depends on the CAN_MODE_1:0 bits. When FS1B is asserted low, the mode of the CAN physical layer can be configured with the FS1B_CAN_IMPACT bit in the Fail-safe logic and the CAN_DIS_CFG bit in the Main logic, as described in Table 39.

Table 39. CAN mode during FS1B assertion

FS1B pin	FS1B_CAN_IMPACT	CAN_DIS_CFG	CAN mode
HIGH	X	X	No change
LOW (asserted)	0	0	
	0	1	
	1	0	RX only
	1	1	Sleep/no wake-up capability

- Internal register:
  In the FS6500 and FS4500, a register can be configured during initialization phase to define the CAN behavior when FS1B is asserted low (in conjunction with **INIT_WU2** register, **CAN_DIS_CFG** bit).
  INIT_FAULT register, **FS1B_CAN_IMPACT** bit.
  By default, the **FS1B_CAN_IMPACT** bit is activated.

Table 40. INIT_FAULT

Write																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	1	1	0	0	1	0	0	P	FLT_ERR_FS	FS1B_CAN_IMPACT	0	FLT_ERR_IMPACT	Secure_3	Secure_2	Secure_1	Secure_0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	SPI_FS_ERR	SPI_FS_CLK	SPI_FS_REQ	SPI_FS_PARITY	FLT_ERR_FS	FS1B_CAN_IMPACT	RESERVED	FLT_ERR_IMPACT

FS1B_CAN_IMPACT	Description	Define CAN behavior when FS1B is asserted low
	0	No effect
	1	CAN in Rx only or sleep mode when FS1B is asserted (depends on the CAN_DIS_CFG bit in INIT_WU2 register)
	Reset condition	Power on reset

- Internal register:
  In the FS6500 and FS4500, a register can be configured during initialization phase to define the CAN behavior when FS1B is asserted low (in conjunction with **INIT_FAULT** register, **FS1B_CAN_IMPACT** bit).
  INIT_WU2 register, **CAN_DIS_CFG** bit.
  By default, the **CAN_DIS_CFG** bit configure the CAN physical layer in RX only mode.

Table 41. INIT_WU2

Write																
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
MOSI	1	0	0	0	0	1	1	P	WU_IO5_1	WU_IO5_0	CAN_DIS_CFG	CAN_WU_TO	0	LIN_J2602_DIS	LIN_SR_1	LIN_SR_0
MISO	SPI_G	WU_G	CAN_G	LIN_G	IO_G	VPRE_G	VCORE_G	VOTHERS_G	WU_IO5_1	WU_IO5_0	CAN_DIS_CFG	CAN_WU_TO	RESERVED	LIN_J2602_DIS	LIN_SR_1	LIN_SR_0

CAN_DIS_CFG	Description	Define CAN behavior when FS1B is asserted low
	0	CAN in Rx only mode (when FS1B_CAN_ IMPACT = 1 in INIT_FAULT register)
	1	CAN in sleep mode (when FS1B_CAN_ IMPACT = 1 in INIT_FAULT register)
	Reset condition	Power on reset

6 Start-up sequence

After power up or after wake-up from LPOFF, the device releases the reset after approximately 16.5 ms. When the reset is released, all the regulators are on and the Fail-safe FS0B and FS1B pins are asserted low (safe state$_{SBC}$). At this stage, both the Main and Fail-safe digitals are in INIT phase. The device is ready to be configured.

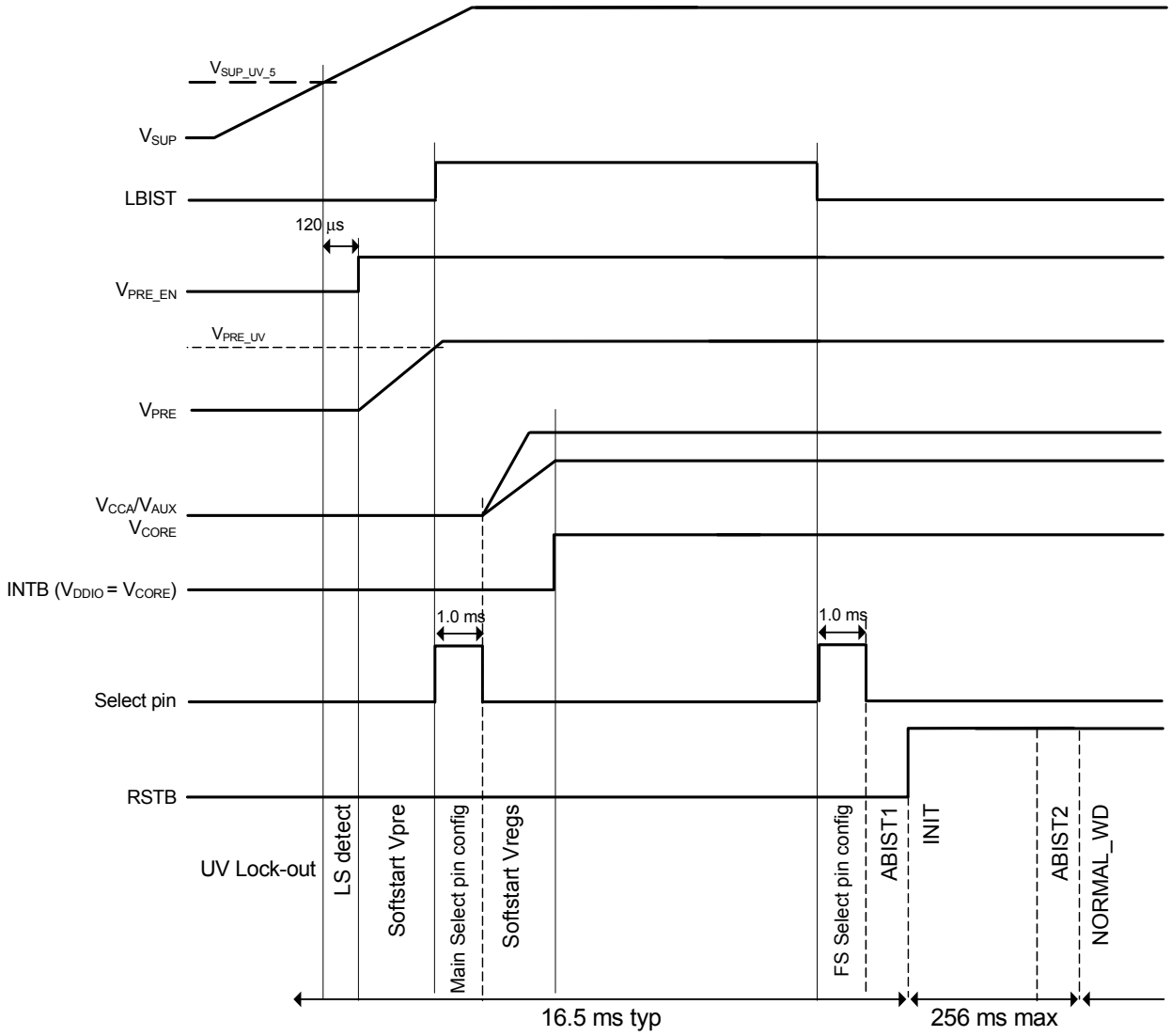


Figure 24. Start-up sequence diagram

When the reset is released, the MCU has an open window of 256 ms to configure the device and to send the first good WD refresh. If the first good WD is not refreshed within 256 ms, a reset is sent to the MCU. After five consecutive resets (because at startup the fault error counter starts at 1), the device moves in deep Fail-safe mode if the SELECT pin is connected to Ground.

6.1 INIT phase

During this phase, all the initialization registers can be accessed and configured. Refer to the FS6500 and FS4500 data sheet to know which registers can be configured during the INIT phase only. Based on the informations in this safety manual, the following is a summary of the safety configuration sequence to do before releasing the safety pins (FS0B and FS1B):

1- Verify

- **Verify** LBIST and ABIST1 are pass.
- **Verify** V_{PRE} buck or buck boost configuration.
- **Verify** V_{CCA} and V_{AUX} voltage configuration (3.3 V or 5.0 V).
- **Verify** debug mode is not activated.
- **Verify** deep Fail-safe configuration.

2- Configure

- **Configure** V_{CORE}, V_{CCA}, and V_{AUX} overvoltage and undervoltage impact on RSTB and FS0B.
- **Configure** V_{AUX} as tracker of V_{CCA} if using V_{AUX} to supply sensors and V_{CCA} as an MCU ADC reference voltage.
- **Configure** the WD window period and the WD counters. Ensure the configuration does not violate the FTTI requirement at the system level.
- **Configure** the WD impact on RSTB and FS0B.
- **Configure** the final and indirectly intermediate value of the fault error counter.
- **Configure** RSTB and FS0B behavior when fault error counter $\geq$ intermediate value
- **Configure** the RSTB pulse duration.
- **Configure** MCU FCCU error monitoring in regards to IO[2] and IO[3] connections.
- **Configure** Ext. IC error monitoring in regards to IO[4] and IO[5] connections.
- **Configure** FS1B mode (T_{DELAY} or T_{DUR}) and its associated timing.
- **Configure** FS1B impact on CAN physical layer.

3- Execute

- **Execute** ABIST2 on FS1B and VAUX by the SPI request from the MCU (device with FS1B and V_{AUX} configured safety critical).
- **Close** the Fail-safe INIT phase by sending the first good WD refresh. The Fail-safe registers are locked, the configuration is protected against bit flip, and the registers can be read only. The FCCU monitoring is active and the WD must be correctly refreshed.
- **Configure** and **Close** the Main INIT phase.
- **Clear** all the flags by reading all diag registers of both main and Fail-safe registers.
- **Clear** the fault error counter to “0” by seven consecutive good WD refresh (if WD refresh counter is configured at “6”). Fault error counter is at “1” at start up or after wake-up from LPOFF if no other faults are reported.
- **Close** the switch S1 when FS1B is used
- **Release** FS0B and FS1B pins. The MCU must maintain its own safety path to low, because the safety path check as not been done yet.
- **Perform** a safety path check for both FS0B and FS1B.

When FS1B is used, the sequence order of the **execution** phase can be adjusted to reduce the application starting time. The switch S1 can be closed earlier, just after ABIST2 on FS1B, to allow the charge of C_{PD} while the fault error counter is cleared.

The FS6500 and FS4500 is now ready. If everything is ok for the MCU, it can release its own safety path and the ECU starts.

7 List of Fail-safe errors and potential cascade effects

Impacts on Fail-safe activation (RSTB and FS0B) depends on the product configuration. FS1B follows the activation of FS0B with a configurable delay or for a configurable duration. Refer to the FS6500 and FS4500 data sheet for FS1B timing details. Table 42 represents the device behavior with default configuration at start up.

Green	No impact. Device behavior is not configurable.
Red	Impact by default. Device behavior is not configurable.
Blue	Device behavior is configurable by the SPI during INIT_FS phase only.

Table 42. List of Fail-safe error handling and potential cascade effect

Error	Description	Main action in case of fault	FLT_ERR _CNT	RSTB	FS0B	Potential cascade effect	RSTB	FS0B
Vpre_OC	Overcurrent	V_{PRE} switched off	No effect	High	High	Undervoltage reported on all regulators	High	Low
Vpre_Ilim	Current limitation	Duty cycle reduction	No effect	High	High	Undervoltage reported on all regulators	High	Low
Vpre_OV	Overvoltage on V_{PRE}	V_{PRE} switched off	+1	Low	Low	All regulators are switched off	-	-
Vpre_UV(1)	Undervoltage on V_{PRE}	V_{PRE} kept on	No effect	High	High	Undervoltage reported on all regulators	High	Low
VPRE_TSD	Thermal shutdown	V_{PRE} switched off	No effect	High	High	Undervoltage reported on all regulators	High	Low
Vcore_OV	Overvoltage on V_{CORE}	V_{CORE} switched off	+1	Low	Low	-	-	-
Vcore_UV	Undervoltage on V_{CORE}	V_{CORE} kept on	+1	High	Low	-	-	-
Vcore_Ilim	Current limitation	Duty cycle reduction	No effect	High	High	Undervoltage on V_{CORE}	High	Low
Vcore_TSD	Thermal shutdown	Vcore switched off	No effect	High	High	Undervoltage on V_{CORE}	High	Low
FCRBM_OV	FB_core Resistor Bridge Monitoring (Vcore_OV)	V_{CORE} switched off	+1	Low	Low			
FCRBM_UV	FB_core Resistor Bridge Monitoring (Vcore_UV)	V_{CORE} kept on	+1	High	Low			
Vcca_OV	Overvoltage on V_{CCA}	V_{CCA} switched off	+1	Low	Low	-	-	-
Vcca_UV	Undervoltage on V_{CCA}	V_{CCA} kept on	+1	High	Low	-	-	-
Vcca_ILIM	Current limitation	-	No effect	High	High	Undervoltage on V_{CCA}	High	Low
Vcca_TSD	Thermal shutdown (internal Pmos)	V_{CCA} switched off	No effect	High	High	Undervoltage on V_{CCA}	High	Low
Vaux_OV	Overvoltage on V_{AUX}	V_{AUX} switched off	+1	Low	Low	-	-	-
Vaux_UV	Undervoltage on V_{AUX}	V_{AUX} kept on	+1	High	Low	-	-	-
Vaux_Ilim	Current limitation		No effect	High	High	Undervoltage on V_{AUX}	High	Low
Vaux_TSD	Thermal shutdown (internal reverse transistor)	V_{AUX} switched off	No effect	High	High	Undervoltage on V_{AUX}	High	Low
Vcan _OV	Overvoltage on V_{CAN}	V_{CAN} switched off and CAN Physical layer off	No effect	High	High	-	-	-
Vcan _UV	Undervoltage on V_{CAN}	CAN physical OFF	No effect	High	High	-	-	-
Vcan_ILIM	Current limitation		No effect	High	High	-	-	-
Vcan_TSD	Thermal shutdown	V_{CAN} switched off and Physical layer OFF	No effect	High	High	-	-	-

Table 42. List of Fail-safe error handling and potential cascade effect (continued)

Error	Description	Main action in case of fault	FLT_ERR _CNT	RSTB	FS0B	Potential cascade effect	RSTB	FS0B
IO_2:3	MCU FCCU error handling	-	+1	Low	Low	-	-	-
IO_4:5	External IC error handling	-	+1	High	Low	-	-	-
WD	Watchdog	WD Not OK during INIT phase	+1	Low	Low(3)	-	-	-
		WD_ERR_CNT = max value in Normal_WD	+1	Low	High	-	-	-
RSTB shorted to High	Short-circuit	-	No effect	High (externally)	Low	FLT_ERR_CNT +1 by FS0B assertion	-	-
FS0B shorted to high(2)	Short-circuit	-	No effect	Low (Normal_WD)	High (externally)	-	-	-
FS1B shorted to high	Short-circuit	-	No effect	High	Low	-	-	-
SPI DED	Dual error detection in SPI		+1	High	Low	-	-	-
LBIST	Logic built-in self test	Keep system in Fail-safe	No effect	High	Low	Deep Fail-safe or LPOFF-Sleep if RSTB = 8.0 s	-	-
ABIST1	Analog built-in self test	Keep system in Fail-safe	+1	High	Low	Deep Fail-safe or LPOFF-Sleep if RSTB = 8.0 s	-	-
ABIST2 (V_{AUX}/FS1B)	Analog built-in self test	Keep system in Fail-safe	+1	High	Low	-	-	-

Note:

1. No action either on RSTB nor on FS0B exists following V_{PRE} undervoltage event. if V_{PRE} is used to supply a function in the ECU, a safety mechanism at application level must be taken in case of undervoltage, if considered as a safety regulator.
2. If FS0B short to high is detected during ABIST1, RSTB is released for MCU diagnostic. If FS0B short to high is detected during normal mode, RSTB is asserted low.
3. FS1B is asserted with T_{DELAY} default value (37 ms).

8 Acronyms and abbreviations

A short list of acronyms and abbreviations used in this document is summarized for completeness:

Table 43. Acronyms and abbreviations

Term	Meaning
ABIST	Analog built-in self-test
ADC	Analog-to-digital converter
BIST	Built-in self-test
CCF	Common cause failure
CF	Cascading failure
CMF	Common mode failure
DPF	Dual-point fault
FCCU	Fault collection and control unit
FCRBM	Feedback core resistor bridge monitoring
FMEDA	Failure modes, effects & diagnostic analysis
FSM	Fail-safe machine
FSO	Fail-safe outputs
FSSM	Fail-safe state machine
FTTI	Single-point fault tolerant time interval
GPIO	General purpose I/O
MPFDI	Multiple-point fault detection Interval
LBIST	Logic built-in self-test
LF	Latent fault
LFSR	Linear feedback shift register
MCU	Microcontroller unit
MPF	Multiple-point fault
OV	Overvoltage
PST	Process safety time
RF	Residual fault
SBC	System basis chip
SF	Safe fault
SPF	Single-point fault
UV	Undervoltage

8.1 Safety tags

Table 44. Safety tags

Tag	Assumption
[SMR_01]	It is assumed the FS6500 and FS4500 are used in “12 volts automotive “applications for which the battery voltage (i.e. pin VSUP1, VSUP,2, VSUP3, and VSENSE) never exceeds the maximum ratings of the FS6500 and FS4500 (i.e 40 V). Above this voltage, the FS6500 and FS4500 run the risk of being destroyed and the safety requirements are no longer satisfied.
[SMR_02]	It is assumed the FS6500 and FS4500 is used in application for which the fault tolerant time interval is ≥ 10 ms. Shorter “fault tolerant time interval” must be deeply analyzed.
[SMR_03]	It is assumed the FS6500 and FS4500 is used in application for which the mission profile is like in Table 3, Temperature profile for mission profiles (or less aggressive).
[SMR_04]	It is assumed when the multiple point fault time interval is ≤12 hours, then the driving cycle is assumed to be ≤ 12 hours.
[SMR_05]	It is assumed the normal operating range of the FS6500 and FS4500 is fulfilled by the compliance to the FS6500 and FS4500 data sheet.
[SMR_06]	To avoid systematic errors during system integration, it is system integrator’s responsibility to follow NXP recommendations as described in the FS6500 and FS4500 data sheet and application note available at www.nxp.com.
[SMR_07]	It is system integrator’s responsibility to report all field failures of the devices to the silicon supplier.
[SMR_08]	It is system integrator’s responsibility to take into account the latest device errata during system design, implementation, and maintenance. For a functional safety-related device such as FS6500 and FS4500, this also concerns functional safety-related activities such as system level functional safety concept development.
[SMR_09]	It is the system integrator’s responsibility to ensure the system transitions itself to a safe $state_{system}$ when the FS6500 and FS4500 explicitly indicates an error via its Fail-safe outputs (FS0B and FS1B).
[SMR_10]	It is the system integrator’s responsibility to ensure the system transitions itself to a safe $state_{system}$ when the FS6500 and FS4500 is completely unpowered.
[SMR_11]	It is assumed single-point and latent fault diagnostic measures complete operations (including fault reaction) in a time shorter than the respective FTTI when the safety function is enabled.
[SMR_12]	It is assumed simultaneous pin disconnections (i.e. pin lift on the PCB) are restricted to 1 during pin FMEA and FMEDA analysis.[
[SMR_13]	It is assumed the thermal connection of the exposed-pad to the PCB is always ensured due to its large size.
[SMR_14]	Short-circuit between PCB tracks is not considered.
[SMR_15]	External component disconnection is not considered.
[SMR_16]	It is the system integrator’s responsibility to make sure the MCU checks for SPI DED error at each WD refresh, by checking the ERR_INT_SW flag when writing in WD_ANSWER register.
[SMR_17]	It is the system integrator’s responsibility to make sure the MCU checks the configuration of the buck/or boost configuration after system startup or after LPOFF mode, by reading the LS_ DETECT bit in HW_CONFIG register.
[SMR_18]	It is system integrator’s responsibility to make sure the MCU checks the VPRE undervoltage flag, in case an external circuitry is connected to VPRE as a supply of this circuitry, by reading the VPRE_UV bit in the DIAG_VPRE register.
[SMR_19]	It is the system integrator’s responsibility to make sure the right resistor values are well connected between VCORE_SNS and ground, with the middle point connected to FB_CORE to configure the right voltage to MCU.
[SMR_20]	It is assumed the value of the external resistor bridge stays within its nominal value.
[SMR_21]	It is the system integrator’s responsibility to make sure the MCU checks the VCCA output voltage level after system startup, or after LPOFF mode by reading the VCCA_HW bit in the HW_CONFIG register.
[SMR_22]	It is the system integrator’s responsibility to take measurements at the system level maintain the safe $state_{system}$ when the V_{CCA} supply voltage is above or below the specified operational range.
[SMR_23]	It is the system integrator’s responsibility to make sure the MCU checks the VAUX output voltage level after system startup or after LPOFF mode by reading the VAUX_HW bit in the HW_CONFIG register.
[SMR_24]	It is the system integrator’s responsibility to take measurements at the system level and maintain the safe $state_{system}$ when the VAUX supply voltage is above or below the specified operational range.
[SMR_25]	It is the system integrator’s responsibility to make sure the V_{CCA} linear regulator is used as reference voltage of the analog to digital converter of the MCU.
[SMR_26]	It is the system integrator’s responsibility to make sure the bi-stable protocol is configured in the NXP MCU for FCCU protocol.
[SMR_27]	It is the system integrator’s responsibility to validate other MCU error monitoring implementation at system level.

Table 44. Safety tags (continued)

[SMR_28]	It is the system integrator's responsibility to make sure the error output signal from the external IC is well connected to the SBC IO[4] and one MCU GPIO, to ensure the MCU is able to listen to the fault.
[SMR_29]	It is the system integrator's responsibility to make sure the MCU refresh periodically the FS6500 and FS4500 watchdog.
[SMR_30]	It is the system integrator's responsibility to make sure the MCU checks for ERR_INT_HW flag at each WD refresh when writing in WD_ANSWER register.
[SMR_31]	It is the system integrator's responsibility to make sure external components connected to RSTB are available to bring the safety critical outputs to known levels during operation.
[SMR_32]	It is the system integrator's responsibility to make sure external components connected to FS0B are available to bring the safety critical outputs to a known level during operation.
[SMR_33]	It is the system integrator's responsibility to make sure opening the safety switch in a system must not be driven by the FS0B only, but also by the MCU or FS1B.
[SMR_34]	It is system integrator's responsibility to make sure a resistor in series is well connected to FS0B.
[SMR_35]	It is the system integrator's responsibility to make sure external components connected to FS1B are available to bring the safety critical outputs to a known level during operation.
[SMR_36]	It is system integrator's responsibility to make sure a resistor in series is well connected to FS1B.
[SMR_37]	It is system integrator's responsibility to make sure R_{PD} and C_{PD} are well connected to VPU_FS.
[SMR_38]	It is the system integrator's responsibility to make sure the MCU checks to see both LBIST and ABIST1 are pass after start-up and resuming from LPOFF mode.
[SMR_39]	It is the system integrator's responsibility to make sure the MCU executes the ABIST2 by the SPI request and checks to see both FS1B and VAUX ABIST are pass by reading ABIST2_FS1B_OK and ABIST2_VAUX_OK bits in the BIST register.
[SMR_40]	It is the system integrator's responsibility to make sure the MCU checks the deep Fail-safe feature is activated after system startup or after LPOFF mode by reading both DFS_HW1 bit in HW_CONFIG register (main logic detection) and DFS_HW2 bit in DEVICE_ID_FS register (Fail-safe logic detection).
[SMR_41]	It is the system integrator's responsibility to make sure the MCU checks the debug mode is not activated after system startup or after LPOFF mode by reading the DBG_HW bit in HW_CONFIG register.

If one of these safety manual requirements is not respected, the impact in the FMEDA metrics must be verified.

9 Document revision history

This document applies for the silicon revision DEV_REV_2:0 = ‘010’ in DEVICE_ID register. Refer to the FS6500 and FS4500 data sheet for more details on DEVICE_ID register read access. Table 45 summarizes the revisions to this document.

Table 45. Revision history

Revision	Date	Description of changes
1.0	6/2016	• Initial release • Applies for the silicon revision DEV_REV_2:0 = ‘010’ in DEVICE_ID register
2.0	6/2016	• Corrected FIT rate number in 3.2, Failure rates, page 14 • Added a note at the end of the Table 44

How to Reach Us:

Home Page:
NXP.com

Web Support:
http://www.nxp.com/support

Document Number: FS6500-FS4500SMUG
Rev. 2.0
6/2016