# Avoiding Functional Safety Compliance Pitfalls in the Motor-Control Design Process

Texas Instruments

**Bharat Rajaram**
Systems Engineering Manager
Arm-Based Microcontrollers

**System design and functional safety compliance should not happen serially. Unfortunately, traditional design approaches – and many organizations – treat these steps in the design process as separate, siloed activities, often leading to increased design costs and delays getting to market.**

## At a glance

### 1 Defining functional safety compliance

The goal of functional safety standards is to manage and mitigate systematic faults while also being able to detect and prevent (or, at minimum, render safe) random hardware failures when they occur.

### 2 Two attributes of functional safety system design

Functional safety involves developing systems to deliver an intended function and to meet a safety integrity level.

### 3 The recommended approach to designing functionally safe motor-control and drive systems

System engineers designing functionally safe systems should approach functional safety compliance at the outset of the design process – not as an afterthought.

When designing functionally safe motor-control applications, should you tackle functional safety compliance at the beginning, as an initial design requirement? Or should you treat functional safety as an add-on feature, incorporated into the final stages of your design?

Functional safety should be part of the initial design requirements – interwoven with the intended functionality of the motor drive. This isn't the norm, because traditional system design workflows don't approach safety compliance synergistically. But neglecting to consider how you need to meet safety integrity compliance at the outset can result in costly delays when introducing systems to market.

The onset of Industry 4.0 and the growth of vehicle electrification and connectivity require that we change our approach to functional safety compliance. Simply put, we now have more motor systems in more applications, and a high bar for complying with functional safety standards.

## Defining functional safety compliance

The goal of functional safety standards such as International Electrotechnical Commission (IEC) 61508 and International Organization for Standardization (ISO) 26262 is to manage and mitigate systematic faults while also being able to detect and prevent (or, at minimum, render safe) random hardware failures when they occur.

The adoption of a rigorous development process with independent verification and validation can help manage for systematic faults.

It is possible to detect, prevent or render safe random hardware failures by:

- Having a thorough understanding of the equipment under control.
- Analyzing the likely sources of situational hazards and their attributes, such as probability of occurrence, severity of impact and controllability of the incident.

The pairing of safety mechanisms with each situational hazard then helps designers meet quantitative metrics such as safe failure fraction (SFF) and probability of failures/hour (PFH) as required by IEC 61508. For example, a Safety Integrity Level (SIL) 2 system must have an SFF≥90% and a PFH of ≤1000 failures in time over 1 billion hours of operation.

## Two attributes of functional safety system design

Functional safety standards assume that all systems will fail (not a matter of if, just a matter of when), and there is no such thing as zero risk.

The two attributes of a functional safety system design are developing a system to deliver an intended function and developing the same system to meet a safety function such as a certain SIL or automotive SIL (ASIL).

Designers often approach the two aspects disparately, or serially. Designing functionally safe systems for high-volume applications while preserving design budget requirements is challenging. Table 1 outlines examples of intended functions and safety functions in control and drive applications.

To better explain this concept, look at the elevator motor example in Table 1.

The intended function of the elevator is to move people up and down based on user input. If you push a button to get to the fifth floor, the elevator should take you there.

The safety functions of the elevator take it a step further, and could include:

- Taking you smoothly from floor to floor.
- Stopping in level with the landing on each floor.
- Applying the brake automatically if the elevator exceeds a safe speed.

| Functional safety application | Intended function example | Safety function example (and corresponding SIL or ASIL target) |
|---|---|---|
| **Industrial: elevator motor** | Move elevator up and down in response to user requests | • Start or stop elevator safely (avoid jerks) (SIL 2)<br>• Apply automatic braking if elevator is traveling too fast (SIL 3) |
| **Automotive: electric vehicle (EV) traction motor** | Move EV forward and backward per driver command through accelerator or brake | • Prevent insufficient or excess torque at acceleration (ASIL C)<br>• Prevent braking too hard (to avoid being rear-ended) (ASIL D) |
| **Industrial: steel press** | Control servo-drive system that operates a steel press without lowering factory productivity | • Safe-torque-off (STO) de-energizes drive controller if over torque or over speed occurs (SIL 3)<br>• Safe-limited-speed (SLS) keeps motor speed within acceptable limits if operator is close (SIL 2)<br>• Trigger STO if SLS exceeds a bounds check (to balance between productivity and safety that drives a higher SIL, for example, SIL-3) |

***Table 1.*** *Intended and safety function examples in control and drive applications.*

To better understand how intended functions and safety functions work together, assume that the elevator in a building with 20 floors has a push-button circuit (see Figure 1) with a fault the elevator motor controller interprets as sending the elevator to the 25th or 30th floor (that is, floors that don't exist in the building). A bounds check would catch the fault before it results in an error, or eventually, a failure. This is the accepted progression in functional safety: "faults" lead to "errors," while some errors can lead to "failures."

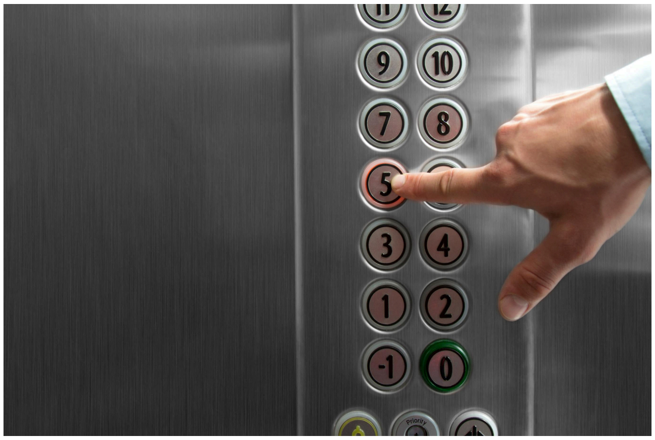

***Figure 1.*** *Example of a modern elevator push button.*

Let's review the processes for an intended function design and a safety function design.

In the intended function design process for a motor drive, a systems engineer selects a microcontroller (MCU) to meet requirements of the intended function. Subsequently, they allocate sense capability, such as integrated analog-to-digital converter (ADC) channels to monitor rotor position, line current, phase voltage and system temperature. The systems engineer then proceeds to use the available processing capabilities of the MCU, such as its CPU's million instructions per second (MIPS) to run the motor-control algorithm, and available actuation peripherals such as pulse-width modulators (PWMs) to drive the motor-driver circuits. This process typically takes several months and also involves designing a printed circuit board (PCB), developing the motor-control algorithms, and developing and debugging all of the embedded software.

In organizations where a separate, somewhat siloed team handles the safety function design process, a separate functional safety expert comes along and reviews the functional safety manual for the MCU that the system engineer originally chose. In some cases, the functional safety expert may discover that the safety-element-out-of-context (SEooC) safety concept calls for the use of SW test of function including error tests, hardware redundancy, a digital-to-analog converter (DAC)-to-ADC loopback check, or monitoring the enhanced PWM through enhanced capture. Recalling the earlier elevator example, it may be necessary to use multiple ADC channels to monitor the level sensor on each floor to protect against a "stuck-at" fault in the MCU's ADC.

If there are insufficient ADC and PWM channels or insufficient CPU MIPS to achieve functional safety, it may be necessary to go back to the drawing board and select a different MCU to realize the functionally safe system – potentially undoing the work completed so far by that separate systems design team.

Even if the design steps do not happen serially, they frequently happen in separate organizational silos; that is, the systems engineer typically does not have any functional safety expertise, and the functional safety expert is not a systems engineer. This siloed approach ultimately results in the same issues: increased system costs and months of time-to-market delays.

## The recommended approach to designing functionally safe motor-control and drive systems

The ultimate goal for system engineers designing functionally safe systems is for them to approach functional safety compliance at the outset of the design process.

Designing and delivering a functionally safe system that meets a design budget requires a synergistic analysis of both safety compliance and intended functionality. Approaching the project independently or serially could result in challenges or even the inability to meet system design goals. Considering the earlier example with the team managing the safety function design process, earlier collaboration would likely have prevented the need to select a new MCU and reconfigure the PCB.

In fact, another example might illustrate the recommended approach. A human brain applies both its left (logical) half and the right (creative) half to holistically solve problems, as shown in **Figure 2**.

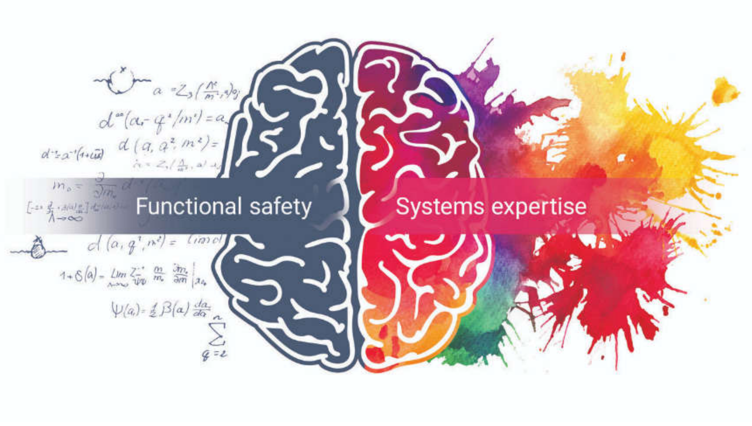

***Figure 2.*** *A single brain has unified expertise in both system design and functional safety compliance.*

Think of the brain as a single organization where each half represents a different team or an internal design resource, capable of bringing their perspective of a particular discipline in the design process. Together, they can work as a single unit in the design workflow, approaching the design from their discipline while staying in clear and constant communication.

Similarly, the most effective design projects use a team of system designers and functional safety experts who work together to realize a functionally safe system.

To help accelerate time to market, systems engineers need the right design resources. For example, TI develops subsystem- and system-level functional safety concepts that are independently assessed by third parties.

## How TI can help you design functionally safe systems

TI's product portfolio ranges from motor drivers and gate drivers to MCUs based on proprietary CPU architectures, including C2000™ and Arm® Cortex®-based MCUs such as the AM2434BSDFHIALVR. These products have advanced diagnostic features and on-chip sensing peripherals that can detect and react to failures quickly while minimizing system downtime (and in an industrial environment, increasing factory productivity).

To help you find the most effective device for a functionally safe design, TI has defined three categories of products suitable for use in functionally safe applications: TI Functional Safety-Capable, TI Functional Safety Quality-Managed and TI Functional Safety-Compliant. (Our motor drivers, gate drivers and MCUs are typically TI Functional Safety-Compliant products.)

TI designs and builds these products to meet the systematic capability compliance recommendations of IEC 61508 and ISO 26262, enabling you to build safe and reliable motor-control and drive systems. We support each device with a failure mode, effects, and diagnostic analysis (FMEDA); a functional safety manual; and (if applicable) a safety diagnostics library, with system and subsystem functional safety concept reports available on TI.com or by request. TI MCU's functional safety manuals include an in-context look at the SEooC and outline possible fault groups for example applications.

Examples of our design resources include a "TÜV SÜD-assessed STO module for industrial drives in the ***TUEV-Assessed Safe Torque Off (STO) Reference Design for Industrial Drives (IEC 61800-5-2)***. Learn more about our functional safety products and view design resources at **www.ti.com/technologies/functional-safety.html**.

TI has experience with ISO 26262 SEooCs and IEC 61508-compliant items, and the types of functionally safe systems TI products are used in. Of course, realizing these benefits requires balancing the complex needs of developing both the intended function and the safety function.

**Important Notice:** The products and services of Texas Instruments Incorporated and its subsidiaries described herein are sold subject to TI's standard terms and conditions of sale. Customers are advised to obtain the most current and complete information about TI products and services before placing orders. TI assumes no liability for applications assistance, customer's applications or product designs, software performance, or infringement of patents. The publication of information regarding any other company's products or services does not constitute TI's approval, warranty or endorsement thereof.

C2000™ is a trademark of Texas Instruments.
Arm® and Cortex® are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.
All trademarks are the property of their respective owners.

# IMPORTANT NOTICE AND DISCLAIMER

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

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

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

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

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

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