# Introduction

This conformity assessment template is for the assessment of safety programmable logic controller (PLC) software / application programs to IEC 61508-3:2010, *Functional safety of electrical/electronic/programmable safety-related systems – Part 3: Software requirements.*

The following notes should be read prior to the assessment:

# General Notes

1. For general guidance on using CASS conformity assessment documents, refer to *The CASS Guide* on using the CASS Methodology available from [www.61508.org/cass](http://www.61508.org/cass) (Document: ‘*CASS-Guide-A’).*
2. Use of this template assumes acceptance of the CASS scheme liability disclaimer in ‘*CASS-Guide-A’*.
3. This conformity assessment template does not replace the standard (IEC 61508-3:2010), it is intended to be used in conjunction with a copy of the standard as a method to manage the assessment of functional safety to support the assessor. The “Purpose of TOE” is a general guide to provide context and scope, and it is the assessor’s responsibility to ensure compliance with all the relevant clauses within the standard.
4. The assessor’s comment section shall be used for positive reporting including reference to the document sections / clauses relevant to evidence compliance.

# Template Specific Notes

1. This assessment template should be used in conjunction with other CASS templates for IEC 61508 (see reference documents below).
2. For every TOE, generally the rigour shall increase with increasing SIL.
3. This conformity assessment template is for the generic PLC software (application program) aspects from IEC 61508-3 for platforms that use ladder logic (LAD), functional block diagram (FBD) or similarly restricted limited variability languages (LVL). The PLC platform must have already been successfully assessed for compliance to the IEC 61508 series of standards.
4. This assessment template is for independently functional safety assessed or certified COTS PLCs with the following characteristics:

* Independently functional safety assessed or certified COTS PLC platform.
* Independently functional safety assessed or certified COTS tools and translators.
* Strongly typed programming language.
* Default support for a modular software approach.
* COTS tools and translators with increased confidence from use (large install base).
* Hardware / firmware monitoring of the PLC cycle time (e.g., using a cyclic behaviour or time triggered architecture).
* Hardware / firmware monitoring of sensors and actuators.
* Diverse redundancy or monitoring techniques for the processing of signals and data.
* Support for software module library functions.
* Compiler or runtime protections for safety-related memory areas.

# References

* CASS-508-FSM – Functional Safety Management (FSM)
* CASS-508-SLC – Safety lifecycle (IEC 61508-1)
* CASS-508-SUB – Subsystem or element (IEC 61508-2)
* CASS-508-SYS – System (IEC 61508-2)
* CASS-508-SW – Software (IEC 61508-3)

# Acronyms

The following acronyms are used in this template:

|  |  |
| --- | --- |
| CASS | Conformity assessment of safety-related systems |
| COTS | Commercially off the shelf |
| FS | Functional safety |
| FSA | Functional safety assessment |
| FSM | Functional safety management |
| LVL | Limited Variability Language |
| PE | Programmable electronic |
| PLC | Programmable logic controller |
| SIL | Safety integrity level |
| SRS | Safety requirements specification |
| TOE | Target of evaluation |
| V&V | Verification & validation |

# Version History

|  |  |  |
| --- | --- | --- |
| Version | Date | Description of change |
| V1 | 27/09/2024 | First issue |
|  |  |  |
|  |  |  |
|  |  |  |

| **TOE Ref.** | **Target of Evaluation (TOE)** | **Purpose of TOE** | **IEC 61508 references** | **Supporting documents** | **Assessor’s comments** |
| --- | --- | --- | --- | --- | --- |
| 1 | Functional safety management (overall) | To ensure the overall FSM approach is in conformance with the IEC 61511 series of standards as relevant for the software.  For all SILs, ensure there is general evidence for conformance with the FSM aspects of IEC 61508-1 in relation to the PLC software. | 1/6.1.1,  1/6.1.2,  1/6.2.3,  1/6.2.12. |  | Delete before use: *This TOE is intended to provide a general status on conformance for the FSM aspects related to software.* |
| 2 | PLC software competence | To ensure that competence management has been applied for the software lifecycle and software development.  For all SILs, ensure there is general evidence of competence management covering the PLC software lifecycle and software development. This includes competence specific to the PLC platform / software and those performing PLC software reviews.  NOTE: This includes competence specific to the safety platform (compared to the “standard” PLC platform) and competence specific to the safety language (e.g. LAD, FBD). | 1/6.2.13. |  | Delete before use: *This TOE is intended to consider only the software aspects of competence.* |
| 3 | PLC software safety planning | To ensure the overall safety planning approach has considered the relevant software aspects.  For all SILs, ensure there is general evidence of safety planning for the PLC software lifecycle and development. | 3/6.2.2. |  | Delete before use: *This TOE is intended to consider only the planning related to the software.* |
| 4 | PLC software configuration management | To ensure the software and associated tools are under configuration management and revision control.  For all SILs, ensure there is evidence of PLC software and tool configuration management and software revision control.  NOTE: This includes software or configurations for sensors and final elements. | 1/6.2.10,  3/6.2.3. |  |  |
| 5 | PLC software modification procedures | To ensure that the PLC software is subject to written modification procedures for controlling and authorizing changes.  For all SILs, ensure there are written modification procedures, a method for identifying / requesting changes, evidence for impact analysis and relevant authorisation before changes.  NOTE: Some safety PLC manufacturers include signature / CRC functionality to support identification of modules and / or software. | 3/7.6.2,  3/7.8.2.1,  3/7.8.2.2. |  |  |
| 6 | PLC backup and restoration procedures | To ensure that the PLC software is subject to written backup and restoration procedures.  For all SILs, ensure there are written backup / restoration procedures for the specific PLC and system relevant for the application. This evidence includes aspects for the backup and restoration of all software and parameters. | None. |  |  |
| 7 | PLC software safety lifecycle | To ensure the PLC software development fits with the overall safety lifecycle and that a specific PLC software safety lifecycle is defined and used.  For all SILs, ensure the PLC software safety lifecycle has each phase defined in terms of its elementary activities, objectives, required input information, output results and verification requirements. A simple V-model like that defined in IEC 62061:2021 is useful. | 3/7.1.1,  3/7.1.2. |  |  |
| 8 | PLC software verification planning | To ensure that the verification of the PLC software is planned.  For all SILs, ensure that the verification of the PLC software (using reviews, analysis, simulation and tests) was planned with written procedures / specifications. | 3/7.9.2.1,  3/7.9.2.2. |  |  |
| 9 | PLC software safety requirements | To ensure the safety requirements are sufficient to design the PLC software and to consider the traceability required from the PLC software.  For all SILs, ensure the PLC software requirements are derived from the system SRS and include:   * individual safety function requirements (including voting), * any additional known requirements (e.g., due to architecture, safety manual limitations / constraints, due to hardware, due to embedded software, due to security). * any requirements rooted in safety planning.   Also, for all SILs, ensure there is general evidence for:   * Semi-formal methods describing the requirements (e.g., logic diagrams, cause & effect diagrams, sequence diagrams). * Forward and backward traceability related to each requirement. * Use of the development tools supplied by the PLC manufacturer. | 3/7.2.1,  3/7.2.2. |  |  |
| 10 | PLC software and diagnostics | To ensure that the PLC software, possibly supported by its platform / subsystem, provides sufficient diagnostics for the system and SIL via monitoring of internal functions (e.g., watchdogs, data validation) or external devices (e.g., sensors and final elements).  For all SILs, ensure that, if relevant, the built-in monitoring and diagnostics of the PLC platform / subsystem are supplemented by designed monitoring and diagnostics software to support the achievement of the SIL.  NOTE: The PLC software safety requirements and PLC design combined should define the overall monitoring and diagnostics strategy. | 3/7.2.2.8,  3/7.4.2.7. |  |  |
| 11 | PLC subsystem selection | To ensure the general approach for PLC software development is appropriate and planned.  For all SILs, ensure that the COTS subsystem / platform for the PLC software is suitable for the application and conforms with IEC 61508 or related standard, as relevant.  NOTE: The COTS subsystem / platform should come with a defined “coding standard” (limitations). If not, a coding standard (rules) will need to be defined. | 2/7.2.3.2,  3/7.4.3.2. |  |  |
| 12 | PLC software tool assessment | To ensure any tool(s) used for the software lifecycle or software development is suitable for its assigned task and does not have a negative impact on the system. Alternatively, to ensure the tool(s) output was confirmed by verification procedures.  For all SILs, ensure there is a relevant positive tool(s) assessment or specific tool(s) output verification.  NOTE: Much of this evidence may come from the PLC manufacturers compliance information. | 3/7.4.4. |  |  |
| 13 | PLC software combining safety functions | To ensure when a single PLC covers various SILs (including non-SIL) the approach is suitable for all SILs (cannot negatively impacted any Safety Function).  For all SILs, ensure that the PLC and software meets the highest overall SIL, has sufficient independence between SIFs, and any SIF cannot negatively impact another SIF. | 3/7.4.2.8 –  3/7.4.2.11. |  |  |
| 14 | PLC security and access control | To ensure that the overall system security risk assessment has been used to influence the PLC configuration.  For all SILs, ensure that the PLC design and coding has considered the relevant security risk and access control requirements. | 1/7.4.2.3,  1/7.5.2.2,  3/D.2.4. |  |  |
| 15 | PLC software safety validation planning | To ensure that the PLC software is part of the overall validation planning including specific PLC aspects.  For all SILs, ensure that the PLC software validation plan covers all the functions, the technical strategy, the procedures to be used, the validation environment and the pass / fail criteria. | 3/7.3.1,  3/7.3.2. |  |  |
| 16a | PLC software design | To ensure PLC software design addresses all safety-related system logic including all process operating modes for each safety function including decomposition into modules if applicable.  For all SILs, ensure that a documented application program design, derived from and traceable to the SRS and PLC software safety requirements, covers all logic, each safety function and each operating mode.  Also, for all SILs, ensure that:   * the safety-related part(s) of the software are kept simple. * all safety-related software is treated as the highest SIL. * the preference is for previously used and proven software modules (libraries). * specific design and development tools have been defined. * the LVL programming language has been defined for each software module. * the Software coding standard has been considered and applied. * the design details checking of data range values e.g., for INT and REAL. * the design details some failure assertion programming e.g., for LAD, 2% asserts. * the design has a general modular approach. * the design planned data / memory tables. * the design details the software architecture. | 3/7.4.1 –  3/7.4.5. |  |  |
| 16b | PLC software design properties | To ensure the PLC software design has appropriate safety-related properties.  Also, for all SILs, ensure that the PLC software design demonstrates:   * completeness with respect to the SRS, * correctness with respect to the SRS, * freedom from ambiguity, and * freedom from design faults. | 3/7.4.3,  3/7.4.5. |  |  |
| 17 | PLC safety communications | To ensure that the communications approach for the PLC subsystem / platform is suitable for the application and SIL(s).  For all SILs, ensure the communication / interface requirements are defined, the communication uses relevant safety techniques, and relevant failure modes have been considered.  NOTE: If the communications PFH / PFD is excluded from the SIF PFH / PFD calculation, check that this is sufficiently insignificant to do so.  NOTE: This TOE may need to consider security aspects for the communication link. | 3/7.2.2.10,  3/7.4.3.2,  2/7.4.11. |  |  |
| 18 | PLC software implementation | To ensure that the PLC software development supports the required safety integrity and is derived from the PLC software safety requirement specification.  For all SILs, ensure that the PLC software development methodology complies with the development tools and restrictions of the system PE subsystem, is produced in a structured manner (e.g., modularity), justifies the use of previously developed libraries, and details clear ownership / identification. To also ensure PLC software implementation is traceable to the PLC software safety requirements.  Also, for all SILs, ensure that the PLC software:   * Is readable, understandable, and testable. * Satisfies all the requirements specified during safety planning. | 3/7.4.6. |  |  |
| 19 | PLC software methodology and tools | To ensure that the PLC software development complies with the constraints of the supplier’s safety manual and that a methodology has been defined to reduce / prevent systematic errors.  For all SILs, ensure that there is evidence for compliance with the PLC platforms safety manual. Also ensure that there is evidence of defined techniques and measures focussed on systematic failures. | 3/7.4.4. |  |  |
| 20 | PLC software verification and testing | To ensure that the PLC software is appropriately verified.  For all SILs, ensure that the PLC software and any decomposition into modules is verified by a combination of analysis, simulation, testing (using written procedures and test specifications). The verification must also ensure that the coding standard has been followed and complied with. To also ensure that the scope of the testing is appropriate for the application.  Also, for all SILs, ensure there is general evidence for:   * Simulation supported verification. * Functional and black box testing. * Traceability of tests to/from the software safety requirements specification. * Static analysis of the code and data. | 3/7.4.8,  3/7.9.1,  3/7.9.2.3 –  3/7.9.2.8,  3/7.9.2.10 –  3/7.9.2.14. |  |  |
| 21 | PLC software review | To ensure that the PLC software is reviewed by a competent person not involved in the development.  For all SILs, ensure that the PLC software review(s) were structured, undertaken and documented. To also ensure that any review(s) was undertaken by a competent person. | 3/7.4.6.1. |  |  |
| 22 | PLC software subsystem compliance | To ensure that the PLC platform is configured and used as per the manufacturer’s requirements and recommendations (e.g., as per any certificate and its safety manual).  For all SILs, ensure that PLC software review specifically considers the requirements and recommendations from the PLC platforms safety manual. | 3/7.4.3.2. |  |  |
| 23 | System integration test | To ensure the PLC software has been successfully integrated onto the target platform / subsystem including interaction with a sample set of field devices and or simulator.  For all SILs, ensure the PLC software integration is performed, based on the initial integration test requirements, with documented test results. | 3/7.5. |  |  |
| 24 | PLC software modification V&V | To ensure that any modifications have been correctly requested, authorised, planned, and delivered including relevant verification or validation.  For all SILs, ensure the modification procedures (incl. impact analysis) have been followed regardless of the lifecycle phase e.g., modification during testing, modification during operation.  Also, for all SILs, ensure there is general evidence, when relevant, for:   * Reverification of changed modules. * Reverification of affected modules. * Revalidation of the whole PLC software. | 3/7.8.2.3 –  3/7.8.2.10. |  |  |
| 25 | PLC software validation considerations | To ensure that the PLC software is a key part of the overall system validation and that PLC competent persons take part in the system validation.  For all SILs, ensure that PLC software portion of the validation is carried out as planned, considering the software safety requirement specification, and carried out by at least one safety PLC competent person. | 3/7.7.1,  3/7.7.2. |  |  |
| 26 | PLC software FS audit | To ensure that software lifecycle and software development activities are subject to FS audit(s).  For all SILs, ensure that relevant aspects of the software lifecycle and software development have been audited in relation to functional safety.  Also, for all SILs, ensure there is general evidence for:   * Use of pre-defined a checklist(s) in the audit process. * Audit outputs used in functional safety assessments. * Defined failure modes for each safety function. | 3/6.2.3. |  |  |
| 27 | PLC software documentation | To ensure that all PLC software documents are available and that they have been validated for accuracy, consistency, and traceability of the safety function(s).  For all SILs, ensure that all the relevant documents are available and have been validated for accuracy, consistency and traceability of the SIF (from the overall design through to the PLC software), including:   * Software safety requirements specification. * PLC platform / subsystem safety manual. * Software verification and validation plans. * Software design information (system / code). * Software coding standard / programming procedures. * Software libraries / pre-used functions list. * Software verification / review records. * Software test procedures. * Software test specifications. * Software testing results. * Software modification information and results. | 3/5,  3/6.2.3,  3/7.4.4.4,  3/7.4.4.13,  3/7.7.2.7,  3/7.8.2.9,  3/7.9.2.2,  3/7.9.2.5,  3/Annex D. |  |  |