BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF CONTROL AND INSTRUMENTATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FUNCTIONAL SAFETY IN THE PROCESS INDUSTRY FROM THE PERSPECTIVE OF A SYSTEM INTEGRATOR

FUNKČNÍ BEZPEČNOST V PROCESNÍM PRŮMYSLU Z POHLEDU SYSTÉMOVÉHO INTEGRÁTORA

BACHELOR'S THESIS

BAKALÁŘSKÁ PRÁCE

AUTHOR
AUTOR PRÁCE

Matěj Malysa

SUPERVISOR
VEDOUCÍ PRÁCE

doc. Ing. Petr Fiedler, Ph.D.

BRNO 2024

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

Bachelor's Thesis

Bachelor's study program **Automation and Measurement**

Department of Control and Instrumentation

Student:	Matěj Malysa	***ID:***	240391
Year of study:	3	***Academic year:***	2023/24

TITLE OF THESIS:

Functional safety in the process industry from the perspective of a system integrator

INSTRUCTION:

The aim of the thesis is to document the implementation of functional safety requirements in an international contracting company.

1) Briefly discuss the background and requirements arising from IEC 61508 and IEC 61511, discuss the differences between these standards.

2) Propose a safety application design according to IEC 61511.

3) Check and verify the functionality of the individual modules and the application itself in simulated and real HW.

4) Document the implementation process in accordance with IEC 61508 / IEC 61511.

5) Discuss the given implementation and key requirements, including the final declaration of compliance.

RECOMMENDED LITERATURE:

1. Příslušné IEC normy

2. David John Smith, Kenneth G. L. Simpson: SThe Safety Critical Systems Handbook

A Straightforward Guide to Functional Safety: IEC 61508 (2010 Edition), IEC 61511 (2015 Edition) and Related Guidance, Elsevier, 2016 (kniha je k dispozici ve fakultní knihovně)

Date of project specification:	5.2.2024	***Deadline for submission:***	22.5.2024

Supervisor: doc. Ing. Petr Fiedler, Ph.D.

Ing. Miroslav Jirgl, Ph.D.
Chair of study program board

WARNING:

The author of the Bachelor's Thesis claims that by creating this thesis he/she did not infringe the rights of third persons and the personal and/or property rights of third persons were not subjected to derogatory treatment. The author is fully aware of the legal consequences of an infringement of provisions as per Section 11 and following of Act No 121/2000 Coll. on copyright and rights related to copyright and on amendments to some other laws (the Copyright Act) in the wording of subsequent directives including the possible criminal consequences as resulting from provisions of Part 2, Chapter VI, Article 4 of Criminal Code 40/2009 Coll.

ABSTRACT

This thesis summarizes the requirements and key concepts of international standards IEC 61508 and IEC 61511, the relationship between them and their differences. Furthermore, this thesis describes the scope of supply from the point of view of an international contracting company. The thesis also describes the development of a safety instrumented system, the execution process of a safety project including documentation and its testing and verification.

KEYWORDS

IEC 61508, IEC 61511, safety instrumented system, safety instrumented function, hazard & risk analysis, safety-related systems, process industry, process automation, functional safety management system

ABSTRAKT

Tato práce se zabýva shrnutím požadavků a klíčových konceptů vyplývajících z mezinárodních standardů IEC 61508 a IEC 61511, vztahem mezi nimi a jejich rozdíly. Dále tato práce popisuje rozsah práce z pohledu mezinárodní dodavatelské společnosti. Práce rovněž popisuje vývoj softwaru bezpečnostně přístrojového systému, proces řízení projektu včetně dokumentace a jeho následné testování a verifikaci.

KLÍČOVÁ SLOVA

IEC 61508, IEC 61511, bezpečnostně přístrojové systémy, bezpečnostně přístrojové funkce, analýza rizik a nebezpečí, systémy související s bezpečností, průmyslové procesy, procesní automatizace, systém řízení funkční bezpečnosti

Typeset by the thesis package, version 4.09; https://latex.fekt.vut.cz/

Rozšířený abstrakt

Funkční bezpečnost se zabývá identifikací rizik a nebezpečí, které by mohly zapříčinit smrt osob, enviromentální nebo věcnou škodu a nalezení tolerovatelné frekvence vzniknutí těchto nebezpečných situací. Cílem funkční bezpečnosti je minimalizace rizika na takovou úroveň, která je všeobecně společensky přijatelná.

V první kapitole se práce věnuje rozboru požadavků a klíčových elementů vyplývajících z mezinárodních norem IEC 61508 a IEC 61511. Důležitými prvky jsou například metody určení požadovaného snížení rizika a jejich klasifikací do čtyř diskrétních úrovní bezpečnosti (SIL).

V této kapitole se práce dále zabývá přehledem záběru norem IEC 61508 a IEC 61511. IEC 61508 je norma určena především pro výrobce zařízení, které musí splňovat jistou míru bezpečnosti, či pro normotvůrce, kteří vyvíjí normy pro konkrétní aplikační oblasti. IEC 61511 je příkladem takovéto oblasti, zabývá se pouze funkční bezpečností z hlediska procesní automatizace.

Společným rysem obou těchto standardů je přístup, který řídí projekt formou životního cyklu od analýzy rizik, návrhu bezpečnostně přístrojového systému, jeho verifikaci a validaci, uvedení do provozu až po jeho vysloužení a demontáž.

Na konci první kapitoly této práce je vymezen rozsah práce mezinárodní dodavatelské společnosti, která v životním cyklu figuruje jako systémový integrátor.

V druhé kapitole jsou v této práci vymezeny požadavky na konkrétní systém, jejich vztah ke standardu IEC 61511 a případné odklony od této normy. Tato kapitola zahrnuje popis bezpečnostně přístrojových funkcí, jejich chování v případě chyby, provozní režimy a požadovaný čas odezvy systému. Dále jsou popsány požadavky na manuální ovládání, resetování funkcí, komunikace s existujícím řídícím systémem a další.

Na začátku třetí kapitoly je uveden příklad hardwaru splňujícího jak požadavky normy, tak tohoto konkrétního projektu.

Následně se práce zabývá vývojem aplikačního programu na základě dodané zákaznické dokumentace a stanovených požadavků na bezpečnostní systém. Kapitola popisuje všechny vyvinuté knihovny a funkční bloky využité v rámci tohoto projektu. Funkční bloky vykonávají funkce jako například zpracování digitálních a analogových vstupů, nastavení digitálních výstupů, nouzové vypnutí nebo funkcionalitu pro režimy s manuálním ovládáním.

Návrh aplikace vychází ze základních principů objektově orientovaného programování, nejvíce je pak kladen důraz na znovupoužitelnost pro budoucí projekty.

Aplikační program je rozčleněn podle produkčních linek, jejichž funkcionalita je určena diagramy příčin a následků. Jednotlivé produkční linky jsou poté sestaveny z knihovních prvků vytvořených v rámci této práce.

Na konci třetí kapitoly, je ukázka princip spojení všech produkčních linek do jedné aplikace, která poběží v bezpečnostní řídící jednotce v továrně u zákazníka.

Ve čtvrté kapitole se práce zabývá metodologií testování, verifikace a validace bezpečnostních systémů, která byla vytvořena na základě normy IEC 61511. V této kapitole je popsán scénář, podle kterého testy běžně probíhají, V-Model určující posloupnost jednotlivých aktivit pro prokazatelnou verifikaci a validaci systému a popis průběhu testů na tomto projektu.

V rámci této kapitoly jsou zmíněny oba pokusy o otestování systému a jejich výsledky. Výsledkem této práce je systém, který byl úspěšně otestován, pouze s drobnými nedostatky, které byly však zákazníkem akceptovány a budou otestovány až při finální validaci přímo v továrně během července 2024.

Tato práce se z velké části opírá o dokumentaci vyvíjenou souběžně s projektem, kde autor této bakalářské práce je spoluautorem, ale jedná se o nezveřejnitelnou interní dokumentaci.

MALYSA, Matěj. *Functional safety in the process industry from the perspective of a system integrator*. Bachelor's Thesis. Brno: Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Control and Instrumentation, 2024. Advised by doc. Ing. Petr Fiedler, Ph.D.

Author's Declaration

Author:	Matěj Malysa
Author's ID:	240391
Paper type:	Bachelor's Thesis
Academic year:	2023/24
Topic:	Functional safety in the process industry from the perspective of a system integrator

I declare that I have written this paper independently, under the guidance of the advisor and using exclusively the technical references and other sources of information cited in the paper and listed in the comprehensive bibliography at the end of the paper.

As the author, I furthermore declare that, with respect to the creation of this paper, I have not infringed any copyright or violated anyone's personal and/or ownership rights. In this context, I am fully aware of the consequences of breaking Regulation § 11 of the Copyright Act No. 121/2000 Coll. of the Czech Republic, as amended, and of any breach of rights related to intellectual property or introduced within amendments to relevant Acts such as the Intellectual Property Act or the Criminal Code, Act No. 40/2009 Coll. of the Czech Republic, Section 2, Head VI, Part 4.

Brno ....

author's signature[*]

[*] The author signs only in the printed version.

ACKNOWLEDGEMENT

I would like to thank the advisor of my thesis, doc. Ing. Petr Fiedler, Ph.D. for his valuable input and guidance.

The greatest thanks definitely belongs to my friend and colleague Martin Mondřík, who is also the software lead engineer on this project, for sharing his extensive knowledge with me, reviewing my thesis and generally introducing me to the intricacies of functional safety.

Last but not least, I would like to thank Lukáš Volák, the SEC team leader, for giving me the opportunity to work on this project and learn along the way.

Contents

List of Figures

List of Tables

Introduction

The objective of this thesis is to expand upon the semestral thesis and document the implementation of a safety instrumented system (SIS) from the perspective of an international contracting company, acting as a system integrator.

This thesis will briefly summarize the requirements of IEC 61508 and IEC 61511 standards, discuss their differences and use cases and show how these standards are used by a system integrator to implement the functional safety requirements of the customer.

Next, this thesis will propose a design for a SIS, that meets the functional safety requirements of the customer and conforms to IEC 61511 in all applicable parts, highlighting the process of developing the system and handling deviations from the standard. The process of testing and verification of the proposed design will be documented and the results together with the final agreement of the customer will allow to begin on-site activities and later the start-up of production.

Lastly, this thesis will attempt to show that the design conforms to the standard, however the project is still ongoing and final declaration of compliance will not be provided due to delays on the project and the Extended Factory Acceptance Test (EFAT), commissioning and startup being pushed back to later this year (summer 2024), which is after the submission deadline.

The terms used throughout this thesis shall be understood as defined by IEC 61508 and/or IEC 61511.

1 Functional safety

The goal of functional safety engineering is to identify specific hazardous failures, that could potentially cause deaths, environmental damage or damage to assets and establishing tolerable frequency for each mode of failure [12]. However it is impossible to eliminate all risk:

> “There is no such thing as zero risk. This is because no physical item has zero failure rate, no human being makes zero errors, and no piece of software design can foresee every operational possibility.” [12]

In essence, this means that functional safety engineering according to both of the standards mentioned above aims to reduce the risk to an acceptable level – setting safety integrity targets.

1.1 Identifying safety targets

There are many types of failures, but they can generally be classified as **random hardware failures** and **systematic failures**. The former type is related to specific components, where past failure rates are used to model and predict future performance. With the latter type, which cannot be associated to a single component and is largely dependent on factors unique to every system, this approach is not viable and future performance cannot be reliably predicted. [12]

Identification of safety targets can be done quantitatively or qualitatively [12]:

- **Quantitatively** – the predicted frequency of random hardware failures is compared to a tolerable risk target and, if necessary, adapted by e.g. adding redundancy or using a more reliable component
- **Qualitatively** – applying defensive measures and design to minimize occurrences of systematic failures according to the severity of the tolerable risk target

1.2 Safety Integrity Levels (SIL)

Both IEC 61508 and IEC 61511 use SIL levels to divide the spectrum of probability/frequency of failure into four discrete levels as shown in table 1.1.

Table 1.1: Safety integrity levels [10]

SIL	PFD_{avg}	$PFH\ [h^{-1}]$
4	$\geq 10^{-5}$ to $< 10^{-4}$	$\geq 10^{-9}$ to $< 10^{-8}$
3	$\geq 10^{-4}$ to $< 10^{-3}$	$\geq 10^{-8}$ to $< 10^{-7}$
2	$\geq 10^{-3}$ to $< 10^{-2}$	$\geq 10^{-7}$ to $< 10^{-6}$
1	$\geq 10^{-2}$ to $< 10^{-1}$	$\geq 10^{-6}$ to $< 10^{-5}$

Each safety function is assigned a SIL level. The target risk corresponds to either the first column for continuous/high demand mode of operation (probability of dangerous failure per hour) or the second column for low demand mode of operation (average probability of failure on demand). For a better idea of what these numbers mean an inverse value of PFD_{avg} is the risk reduction factor (RRF). [10]

For example a SIL2 safety function has a maximal RRF of:

$$RRF = \frac{1}{PFD_{avg}} = \frac{1}{10^{-3}} = 1000 \tag{1.1}$$

There are numerous method to assess the hazards in a system that can be used for random hardware failures and/or systematic failures, some of which include [10]:

- **ALARP method** – SIL level is increased until it can be shown that the cost of implementing further risk reduction measures outweighs the benefit of further reducing risk
- **Quantitative method** – usually uses fault trees
- **Risk graph method** – mostly used in machinery sector
- **Layer of protection analysis (LOPA)** – combination of protection layers, which are used to reduce the frequency of demand causes gives a total numeric value

1.3 Overview of IEC 61508

1.3.1 Scope of IEC 61508

IEC 61508 is the generic standard for all safety-related electrical/electronic/programmable electronic (E/E/PE) systems. The main objective for this standard is to provide guidance for developing products and standards for specific application sectors. The secondary goal is to enable development of E/E/PE safety-related systems, if there is no pre-existing application sector product or standard. [9]

IEC 61508 specifies the range and extent of measures and techniques used to avoid and control both software and hardware faults applied during the life-cycle of the project, as well as the functional safety management of the whole project.

Another requirement is to determine the necessary safety integrity level of safety functions using the probability of dangerous failure (on demand/per hour) using reliability modelling techniques.

The requirements for hardware that carries the safety function is determined using a combination of hardware fault tolerance and safe failure fraction of the subsystem to obtain the maximum allowable safety integrity level (SIL).

Safety integrity levels are used to categorize the necessary risk reduction (safety integrity) of each subsystem to into four levels, with SIL1 being the lowest level and SIL4 being the highest. [9]

1.3.2 Conformance to IEC 61508

To meet the criteria stated by this standard, it must be shown that all objectives in relevant clauses have been met. Low complexity E/E/PE safety-related systems may be exempt from some requirements, if there is sufficient field experience to provide confidence that safety integrity can be achieved. [9]

1.4 Overview of IEC 61511

1.4.1 Scope of IEC 61511

IEC 61511 is based on IEC 61508 and is an example of an application sector. Process industries use SISs to implement safety instrumented functions (SIFs).

IEC 61511 is primarily intended to be used with E/E/PE logic solvers and also addresses the instruments (sensors and final elements). [10]

The two core concepts of IEC 61511 are:

- SIS Safety life-cycle
- Safety integrity levels (SIL)

In the majority of situations it is preferable that safety is best achieved by inherently safe process design, however this may not always be possible or practical. In those cases, it is necessary to implement other protective systems.

The approach to implementing IEC 61511 can be summarized as follows:

- Hazard & risk analysis, to identify overall safety requirements
- Allocation of safety requirements to SIS(s)
- Framework applicable to all instrumented functions that are implemented to achieve functional safety

- Functional safety management

1.4.2 Conformance to IEC 61511

To conform to this standard, it is necessary to demonstrate that all requirements in clauses 5 through 19 have been met.

Instruments capable of achieving a certain standard (e.g. products certified according to IEC 61508 or components able to be certified based on prior use) are necessary to ensure effective instrumentation. [10]

1.5 Relationship between IEC 61508 and IEC 61511

These standards are very similar in a lot of aspects, simply due to the fact that IEC 61511 is derived from IEC 61508 and is an example of an application sector. One of the key differences is the intended audience for these standards, IEC 61508 is mainly for suppliers and manufacturers, whereas IEC 61511 is used more by SIS designers and integrators. [10]

The use cases are well illustrated by the following figure:

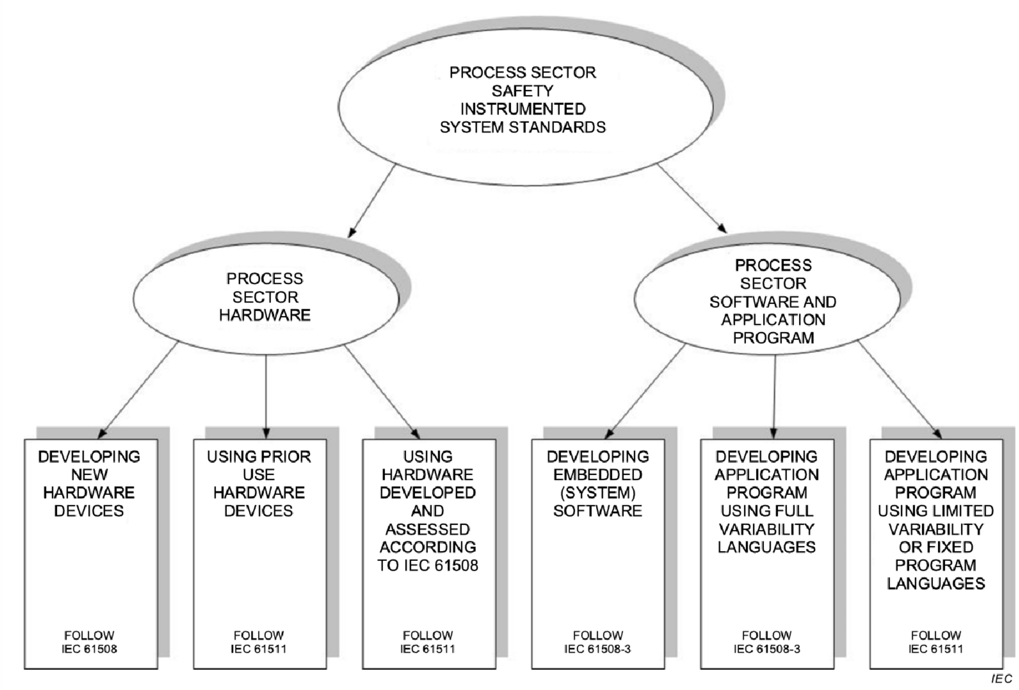

PROCESS SECTOR SAFETY INSTRUMENTED SYSTEM STANDARDS
PROCESS SECTOR HARDWARE
PROCESS SECTOR SOFTWARE AND APPLICATION PROGRAM
DEVELOPING NEW HARDWARE DEVICES
FOLLOW IEC 61508
USING PRIOR USE HARDWARE DEVICES
FOLLOW IEC 61511
USING HARDWARE DEVELOPED AND ASSESSED ACCORDING TO IEC 61508
FOLLOW IEC 61511
DEVELOPING EMBEDDED (SYSTEM) SOFTWARE
FOLLOW IEC 61508-3
DEVELOPING APPLICATION PROGRAM USING FULL VARIABILITY LANGUAGES
FOLLOW IEC 61508-3
DEVELOPING APPLICATION PROGRAM USING LIMITED VARIABILITY OR FIXED PROGRAM LANGUAGES
FOLLOW IEC 61511
IEC

Fig. 1.1: Detailed relationship between IEC 61511 and IEC 61508 [10]

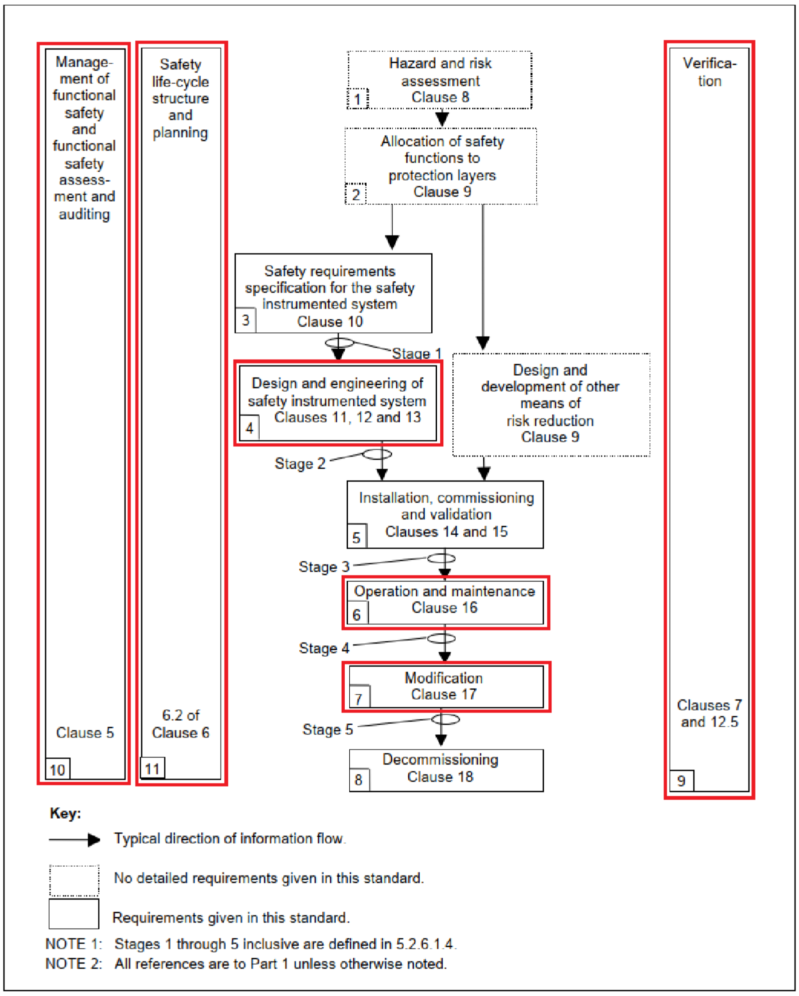
Therefore, for the purpose of this thesis the relationship can be summed up by considering IEC 61511 to be a sector specific adaptation of IEC 61508.

1.6 Implementation from the perspective of a system integrator

1.6.1 Life-cycle approach

To comply with IEC 61511, the safety system shall be implemented using a life-cycle approach, in the context of the project this thesis is based on, the requirements of IEC 61511 clause 5 and 6.2 have been used to develop a functional safety management system (FSMS). The FSMS is an extension of an ISO 9001 quality management system and certified by a 3rd party (TÜV Rheinland) [6].

The relevant parts of IEC 61511, used for the development of the SIS as a system integrator are highlighted in the following figure:



Management of functional safety and functional safety assessment and auditing
Clause 5
10
Safety life-cycle structure and planning
6.2 of Clause 6
11
Hazard and risk assessment Clause 8
1
Allocation of safety functions to protection layers Clause 9
2
Safety requirements specification for the safety instrumented system Clause 10
3
Stage 1
Design and engineering of safety instrumented system Clauses 11, 12 and 13
4
Design and development of other means of risk reduction Clause 9
Stage 2
Installation, commissioning and validation Clauses 14 and 15
5
Stage 3
Operation and maintenance Clause 16
6
Stage 4
Modification Clause 17
7
Stage 5
Decommissioning Clause 18
8
Verification
Clauses 7 and 12.5
9
Key:
Typical direction of information flow
No detailed requirements given in this standard.
Requirements given in this standard.
NOTE 1: Stages 1 through 5 inclusive are defined in 5.2.6.1.4.
NOTE 2: All references are to Part 1 unless otherwise noted.

Fig. 1.2: Stages of SIS safety life-cycle [10]

1.6.2 Scope of supply

The system integrator is mainly involved in phase 4 of the life-cycle and the engineering is limited to only the logic solver subsystem for both hardware and software. This is also applicable to the project this thesis is based on, however the hardware was ordered as a different project and is not in the scope of this thesis. The contents of this subsection are based on the FSM Guidance developed by ABB [5].

Phase 4

The entirety of phase 4 is covered by the development team and includes the complete engineering and documentation of the SIS design according to the specifications provided by the customer as an output of phase 3 performed by either the end-user or an independent company.

Phase 5

This phase is typically out of scope for ABB SEC, but on this specific project the safety team is set to support the customer and ABB Denmark, which is the supplier of the complete project, in on-site activities during the commissioning and validation of the SIS.

Phase 6

Operation & maintenance will be supported by an Operation manual and a Maintenance manual developed by ABB, but ultimately after the handover, the customer will be responsible for operating and maintaining the system.

Phases 7 and 8

These phases are not relevant at this time, since the project is considered as a greenfield.

Any additional customer changes can be considered to be a part of phase 7, but since the project is still ongoing, the modification process has not been started. Instead, the safety requirements checklist (SRCL) is being updated alongside the project.

Phase 9

Phase 9 runs throughout the entire life-cycle of the project and includes reviews of all documents beginning with safety project execution plan, project organogram, safety requirements checklist (SRCL), safety functional design specification (SFDS), test specification & records (TS&R) and all the way until operation & maintenance manual and declaration of conformity.

The EFAT is performed on-site with the customer, but is limited only to the logic solver subsystem and is mostly intended as an extension of phase 4 activities to verify correct integration of software and hardware. The other verification activities such as reviews, testing and integration testing conform to both IEC 61508 and IEC 61511 and maintain an appropriate level of independence.

Phases 10 and 11

ABB uses certified FSMS templates to satisfy the requirements of IEC 61511. The key element of the functional safety management methodology used by ABB is to have competency assessments [1] to ensure sufficient experience and knowledge to execute safety projects. From the design point of view, the development of a SFDS to serve as the basis for the design and specification of the acceptance criteria for testing is the most important part. The people involved in any part of the safety project life-cycle are assessed by a senior competent person to ensure their sufficient qualification and knowledge for any activity.

2 Safety requirements specification

The end-user is responsible for delivering a set of documents – Safety requirements specifications (SRSs), that have been developed according to clause 10.3 of IEC 61511-1. In this specific project the end-user did not deliver these documents and it has been addressed in the Safety Project Execution Plan and the Safety Requirements Checklist document contains cross references to single input documents provided by the end-user instead.

The SRCL document summarizes the safety requirements for the SIS design and includes guidance for the customer on how to satisfy the minimal requirements to ensure that functional safety will be achieved despite the deviations arising from customer documentation and/or demands.

The SRCL document has been updated multiple times during the development of the SIS due to a subpar quality of input information from the customer, which has been improved according to the comments in SRCL, that assess the quality of the input documentation, and numerous change requests that significantly increased cost and development time.

The contents of this chapter are a summary of information from the customer input documentation and the SRCL and SFDS documents [3] [2].

2.1 Definition of SIFs

The first and probably the most notable deviation from the standard is that the end-user has not defined any individual SIFs and instead provided the development team with Cause & Effect (C&E) matrices, which serve as a logic narrative and define the functional relationship between inputs and outputs of the SIS. Another deliverable from the consumer is an IO list containing tag names, trip points, safe states for all signals.

All functions performed by the SIS are referred to as SIFs in the developed documentation for the SIS to avoid confusion when other requirements are concerned and this thesis will refer to them in the same way.

2.2 Behaviour of SIFs

2.2.1 Hardware faults

In case of a channel error/fault of an input card the associated SIF(s) are to be tripped (perform the action to maintain or return to a safe state) and the outputs are set to a safe state. Output failures are covered by a hard wired layer of protection independent of the SIS. If the error/fault affects an entire IO card, the customer has requested to delay the trip for the duration of MTTR (Mean Time To Restoration), with an alarm to indicate a fault/error. The MTTR period starts when a failure is detected by an operator or the SIS and ends when full functionality is restored. This delay is not applied to any critical functions, such as the emergency stop.

2.2.2 Mode of operation

All SIFs have been identified as SIL1 operating in low demand mode. The SIFs operate as normally energized, which acts as de-energize to trip, with the exception of process phase switches and the force enable hardware key, which are normally de-energized and act as energize to trip.

2.2.3 Demanded response time

All SIFs (logic solver subsystem including IO card terminals only) have a demanded response time of less than 4 seconds, which is an ABB assumption based on similar projects, that has been agreed to by the end-user.

2.3 Overrides

2.3.1 Forcing input and outputs

The application is designed to allow up to two forced signals at a time. The access to forcing is limited by including a hardware key that has to be turned to enable the functionality. Hardware key creates another layer of protection from overriding the SIS.

2.3.2 Process phases

The end-user has included two process phases that are activated by physical switches, which effectively disable the SIS. This drastically reduces process safety and ABB recommended that procedures must be developed to ensure, that functional safety is achieved by other risk reduction measures, while the overrides are active. The first process phase is used while the vessels are cooked out and the second is used while the production line is inactive.

2.4 Resetting SIFs

The code is divided by individual C&E matrices and a reset is implemented as a software button on an HMI (Human Machine Interface) display for each C&E related to a production line. Resetting is only allowed if a SIF has returned to a healthy state.

2.5 Interface between SIS and BPCS

The interface between the SIS and BPCS (Basic Process Control System) is to be realized via MasterBus 300 communication. First a MMS (Manufacturing Message Specification) communication is used to transfer data from the SIL application to the non-SIL application in the safety logic solver. This is done because the MasterBus 300 communication cannot be used in a SIL application. The data is then transferred from the non-SIL application by MasterBus 300 communication to an existing Advant 450 controller in the BPCS.

2.6 Other requirements

Other requirements for the SIS are based on industry best practices or have not been developed by the end-user yet. These include e.g. proof testing requirements, startup overrides and others. They are part of the SRCL document as assumptions from ABB side.

3 Safety instrumented system development

3.1 Hardware design

The main focus of this thesis is on the application design, but the hardware for the logic solver subsystem has also been designed by ABB. The most important factor for selecting hardware used in a SIS is the SIL capability. Components such as the PM867 High Integrity controller, SM812 safety module and S800 IO cards are certified for up to SIL3, which exceeds the requirements for this project. ABB 800xA is a complete DCS (Distributed Control System) system programmed using IEC 61131-3 limited variability languages. [7]

(a) PM867 logic solver

(b) AI880A analog input card

Fig. 3.1: Examples of hardware used in this project [8]

3.2 Application software development

The application is developed in Control Builder M (CBM), which supports all IEC 61131-3 languages and also some OOP concepts and Graphic Builder for the creation of HMI displays.

The content of this chapter from this section onward is based on the information in the SFDS [2]

3.2.1 Project structure

In CBM, all projects are divided into libraries, applications and controllers:

```
Project
├── Libraries
│   ├── Standard libraries
│   ├── IACLib
│   ├── SupportLib
│   ├── TypicalLib
│   └── UnitLib
├── Applications
│   ├── App_Sim
│   ├── App_NonSIL
│   ├── App_SIL2
│   ├── VMT
│   └── CTA
└── Controllers
    └── PM867_PSD_01
```

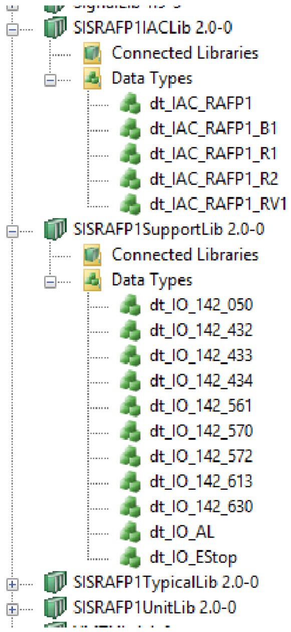
Libraries

Function blocks and control modules from standard libraries are certified for use in SIL applications and provide low-level control modules and function blocks used for hardware access, communication interfaces, data types and other basic functionality such as "logic gates".

Custom libraries have been developed specifically for this project and future projects with the same customer.

IACLib contains custom data types for the SIS to BPCS communication interface. Custom data types are a combination of more primitive data types to reduce clutter in the code and streamline the development process as well as allow quick extensibility in case of project queries.

SupportLib serves the same purpose as IACLib, but the data types are related to individual code blocks (representing C&E matrices) and make the organization of IO signal connections to code blocks easier.



- SISRAFP1IACLib 2.0-0
  - Connected Libraries
  - Data Types
    - dt_IAC_RAFP1
    - dt_IAC_RAFP1_B1
    - dt_IAC_RAFP1_R1
    - dt_IAC_RAFP1_R2
    - dt_IAC_RAFP1_RV1
- SISRAFP1SupportLib 2.0-0
  - Connected Libraries
  - Data Types
    - dt_IO_142_050
    - dt_IO_142_432
    - dt_IO_142_433
    - dt_IO_142_434
    - dt_IO_142_561
    - dt_IO_142_570
    - dt_IO_142_572
    - dt_IO_142_613
    - dt_IO_142_630
    - dt_IO_AL
    - dt_IO_EStop
- SISRAFP1TypicalLib 2.0-0
- SISRAFP1UnitLib 2.0-0

Fig. 3.2: Custom data types in IACLib and SupportLib

TypicalLib contains custom functional block diagrams designed to handle common logic in repeated blocks of code similarly to classes in object oriented languages. The individual typicals developed for this project will be shown in depth in the next section.

UnitLib contains implementation of C&E matrices into functional block diagrams and is a single-purpose library for this specific project. An example of a unit will be shown in section [ref unit design].

Applications

Applications are the equivalent of executables in classic programming and instantiate the code blocks from libraries and connect them together. Every application is assigned to a task in the controller which handles the priority and time of execution.

Simulation application is used only for testing with simulated hardware during software IAT and software FAT, which are done remotely and is not part of the final delivered software to the customer. The control modules in this application use controller access variables to modify input values to the SIL application when downloaded and online in a simulated controller.

Non-SIL application handles the data from SIL application and transfers them into MasterBus 300 communication to BPCS. This application also handles diagnostics, which are generated automatically.

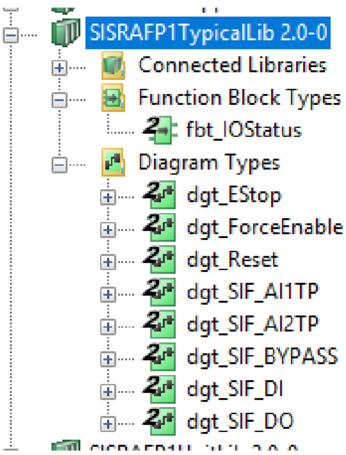
SIL application Control Builder handles SIL1 and SIL2 together, which means that the application is capable of achieving SIL2, but the SIL levels of SIFs are still SIL1 because of field elements capable of maximally SIL1. The SIL application instantiates all diagrams performing the safety functions and connects them together.

Controllers

The last part of the project is related to the hardware configuration, such as tasks, controller IP addresses, IO card settings and signal allocation. The MasterBus 300 communication card is also configured here.

3.3 Typical design

The typicals in this project are programmed almost exclusively using FBD (Functional Block Diagram) programming language with some auxiliary code written in structured text. Typicals are all placed in TypicalLib, which was developed to be used in future projects.



SISRAFP1TypicalLib 2.0-0
Connected Libraries
Function Block Types
fbt_IOStatus
Diagram Types
dgt_EStop
dgt_ForceEnable
dgt_Reset
dgt_SIF_AI1TP
dgt_SIF_AI2TP
dgt_SIF_BYPASS
dgt_SIF_DI
dgt_SIF_DO

Fig. 3.3: TypicalLib

3.3.1 IO Status

This function block merely transforms a `dword` status code to separate boolean values to indicate the status of the IO signal.

3.3.2 Digital input

Digital input contains a signal conditioning block from the standard library and represents a 1oo1 voting architecture. The input is a `BoolIO` parameter, which carries information about the signal value, forcing status and also the value of the IO channel, if force is applied. The `VotedConnection` output transfers a trip command from the typical into the application code.

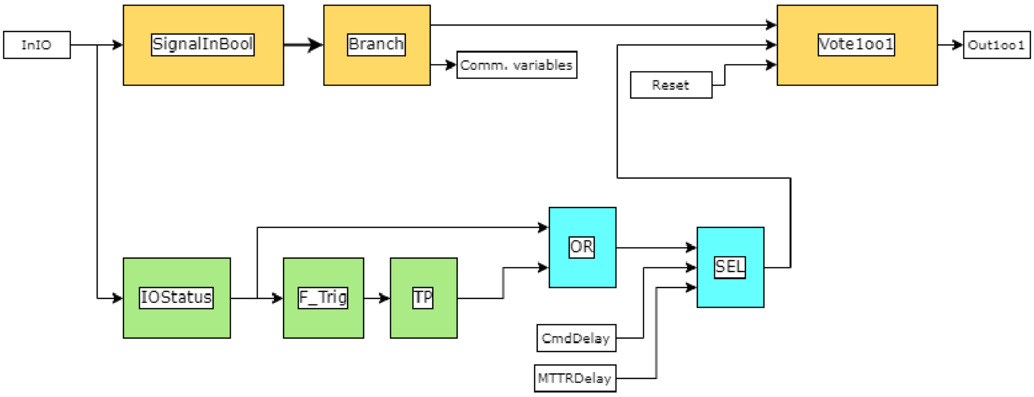


Fig. 3.4: Digital input typical

The auxiliary logic suppresses trips for 72 hours (duration of MTTR) while an IO card error indicated by one of the boolean outputs of the IO status function block is detected and handles overriding during hot replacement of the IO card. The auxiliary logic also handles pre-alarm and alarm for the MTTR delay, which increases availability of the system. Lastly, it provides outputs to the communication interface with the BPCS.

3.3.3 Analog input

The analog input typical has the same functionality as the digital input typical, but handles `RealIO` values. The analog input includes an alarm (High/Low level) and a trip point (High High/Low Low) level, which are externally configurable parameters. The auxiliary logic for trip suppression during IO card and transfer of data to BPCS is the same.

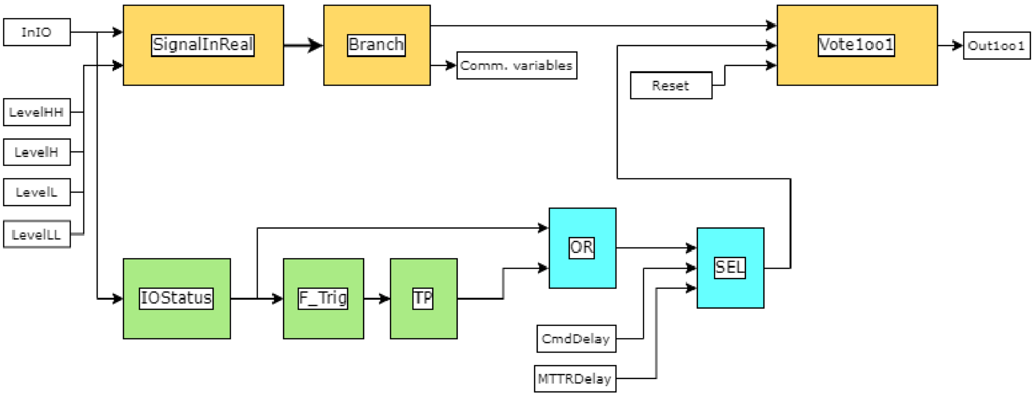


Fig. 3.5: Analog input typical - 1 trip point

It was necessary to also develop a second typical with another voting block for signals, that have both a High and Low trip point active at once.

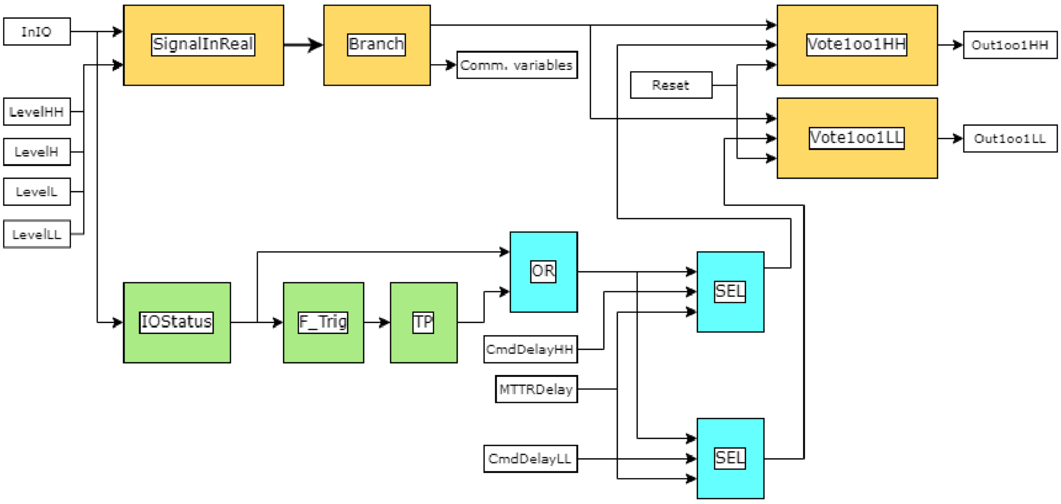


Fig. 3.6: Analog input typical - 2 trip points

3.3.4 Bypass

The bypass typical is used for process phase switches' signals and is very similar to a digital input, but is configured as normally de-energized. Any IO fault will result in inhibition of the bypass command.

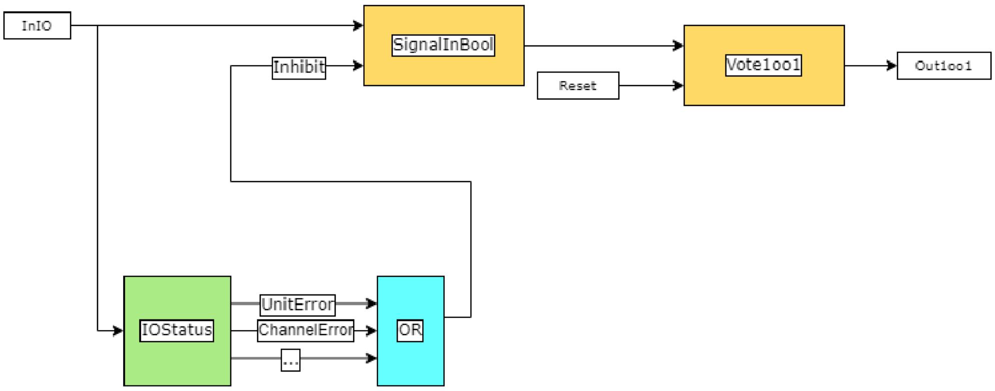


Fig. 3.7: Bypass typical

3.3.5 Force enable

The force enable typical is a very minimalist implementation of a bypass typical with an inverted output, that does not directly enable forcing but instead uses a native function to cancel all forces when the hardware key is not turned. It was developed as a part of the latest change request and had to be added to an already developed system without the modification of other typicals.

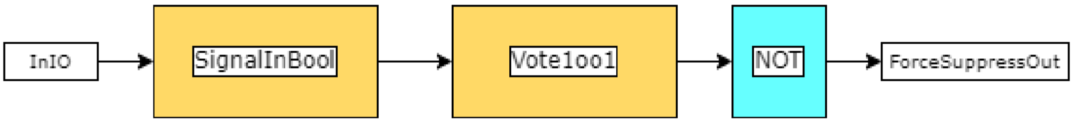


Fig. 3.8: Force enable typical

3.3.6 E-Stop

The E-Stop typical is the only safety function in the entire system to have a 1oo2 configuration. Otherwise, the typical is very similar to a digital input typical without the suppress during IO card fault.

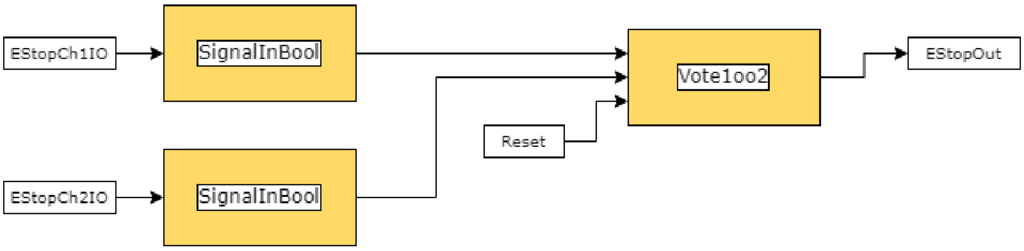


Fig. 3.9: E-Stop typical

This typical was originally developed as a placeholder with dummy signals as E-Stop was not part of the initial project scope and was finished quite late in the project as a part of the latest change request.

3.3.7 Digital output

The digital output processes the trip command and sets the connected output channel to a safe state. The typical reacts to both suppress and trip commands, where the first of these is prioritized – a suppressed output will not trip and a tripped output cannot be suppressed. Both of these commands are overridden by the E-Stop command, which has the highest priority. The typical is also latched in case of an emergency stop, so that the E-Stop must first be returned to a healthy state and only then can the output be reset to a healthy state.

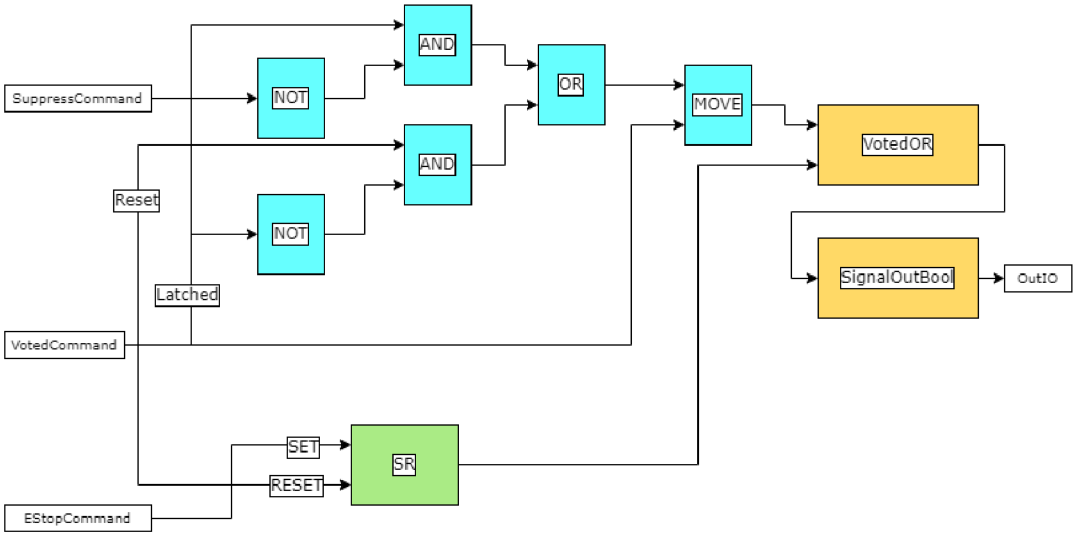


Fig. 3.10: Digital output typical

3.3.8 Reset

This typical is the only one connected to a software signal on an HMI display. The graphic button on the HMI can only write a `True` value to the variable. This typical is used to create a 2 second delay before returing the value back to `False`.

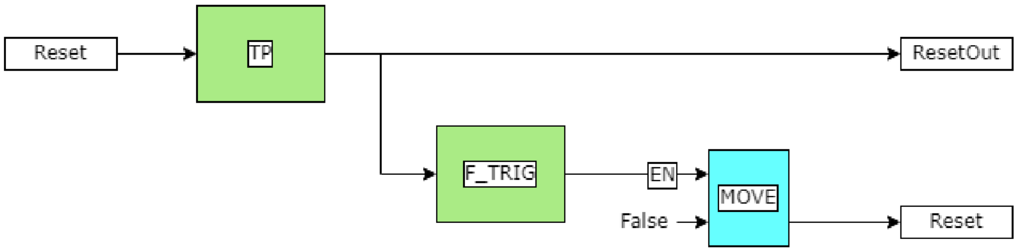


Fig. 3.11: Reset typical

3.4 Unit design

3.4.1 Functional relationships in units

The units in this project are based on delivered C&E matrices from the customer, an example of this matrix is shown in the following two tables, which have been translated from Danish. Rows represent causes and columns represent effects. The markings mean the following:

- **H** – output is set if input reaches high trip point level (indicated by IO list)
- **L** – output is set if input reaches low trip point level (indicated by IO list)
- **0** – output is set if input is de-energized
- **1** – output is set if input is energized
- **>X, &** – signals marked with >X are combined to form signal X, which is marked by tripping levels with & relation, used only for pump protection
- Δ – output is suppressed, when input is energized

The delay indicates trip delay for the combined signals, which is used for pump protection during start up.

Table 3.1: Plant 73 C&E matrix 1/2

Plant 73			MP2 intake safety valve (NC)	Steam to vessel safety valve (NC)	Emergency coolant vessel safety valve (NO)	Emergency coolant column safety valve (NO)	Steam column safety valve (NC)	Steam splash guard safety valve (NC)	vessel circuit safety contactor	Column circuit safety contactor	Column pump safety contactor	Residue pump safety contactor	Output pump safety contactor
Index	Tagname	Delay [s]	1210.V24ZO	1210.V25ZO	1220.V23ZO	1310.V24ZO	1330.V23ZO	1410.V24ZO	P1230.ZO	P1320.ZO	P1360.ZO	P1450.ZO	P1530.ZO
1	TISA021220.15		H	H	H	H	H	H					
2	PISA021220.07		H	H	H	H	H	H					
3	TISA021230.01								H				
4	P021230.ID (P102)								>10				
5	PSA021230.03 (P102)								0				
6	LSA021230.04 (P102)								0				
7	TISA021230.06 (P102)								H				
8	LSA021230.07 (P102)								0				
9	FISA021230.11								>10				
10	SW signal 4+9	60							0&L				
11	TISA021310.16		H	H	H	H	H	H					
12	PISA021310.07		H	H	H	H	H	H					
13	P021320.ID (P113)									>15			
14	FISA021320.11									>15			
15	SW signal 13+14	60								0&L			
16	TISA021360.01										H		
17	TISA021420.15		H	H	H	H	H	H					
18	PISA021420.07		H	H	H	H	H	H					
19	LSA021420.01		0	0			0	0			0		

Table 3.2: Plant 73 C&E matrix 2/2

Plant 73			MP2 intake safety valve (NC)	Steam to vessel safety valve (NC)	Emergency coolant vessel safety valve (NO)	Emergency coolant column safety valve (NO)	Steam column safety valve (NC)	Steam splash guard safety valve (NC)	vessel circuit safety contactor	Column circuit safety contactor	Column pump safety contactor	Residue pump safety contactor	Output pump safety contactor
Index	Tagname	Delay [s]	1210.V24ZO	1210.V25ZO	1220.V23ZO	1310.V24ZO	1330.V23ZO	1410.V24ZO	P1230.ZO	P1320.ZO	P1360.ZO	P1450.ZO	P1530.ZO
20	TISA021450.01 (P117)											H	
21	P021450.ID (P117)											>27	
22	PSA021450.03 (P117)											0	
23	LSA021450.04 (P117)											0	
24	TISA021450.06 (P117)											H	
25	LSA021450.07 (P117)											0	
26	FISA021450.11 (P117)											>27	
27	SW signal 21+26	60										0&L	
28	TISA021530.01 (P118)												H
29	P021530.ID (P118)												>35
30	PSA021530.03 (P118)												0
31	LSA021530.04 (P118)												0
32	TISA021530.06 (P118)												H
33	LSA021530.07 (P118)												0
34	FIQSA021530.11 (P118)												>35
35	SW signal 29+34	60											0&L
36	HS020000.01		0	0	0	0	0	0	0	0	0	0	0
37	HS020000.02		0	0	0	0	0	0	0	0	0	0	0
38	HS021000.01		1	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
39	HS021000.02				Δ	Δ							

3.4.2 Unit implementation

The logic is split into pages according to devices in the unit, the pagination also defines the order of execution of instantiated code blocks. The following figures are a simplified representation of the implemented functional block diagram code for one of the units. The units have input and output parameters connected to the hardware channels via variables with data types from SupportLib.

Process phases switches

The process phase hand switch (HS) signals are connected to instances of bypass typicals and their output bool variables are connected to DO typicals as a direct bypass. HS021000.01 also has a VotedConnection output connected to the trip command of output 1210.V24ZO.

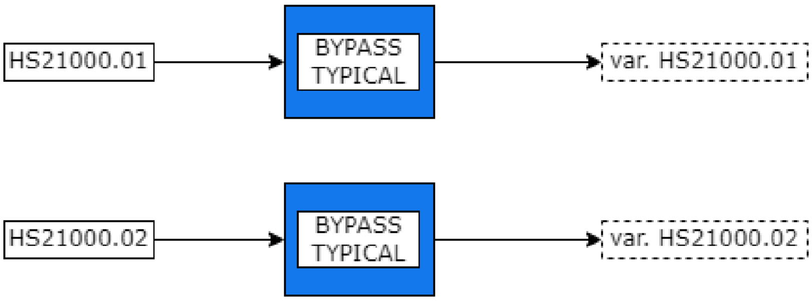


Fig. 3.12: Plant 73 - Process phase switches

Simple inputs

Inputs (causes) are instantiated from the TypicalLib diagrams and configured with correct alarm and trip points or normally energized/de-energized for analog and digital inputs respectively. The inputs are also configure to be latched, until the C&E logic is reset.

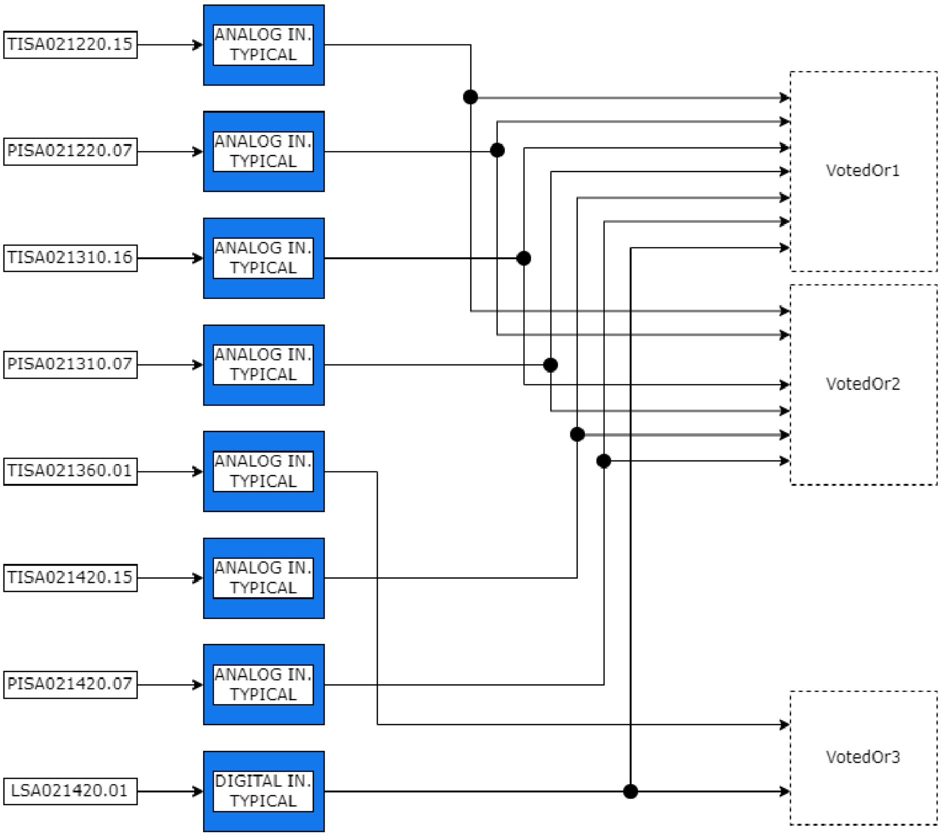


Fig. 3.13: Plant 73 - Simple inputs

Pump protection

Pump protections are logically the most complex parts of each unit, because of the combined software signals that can be seen in tables [ref] and [ref] above. The tripping software signals are created directly in the code and incorporate a delay to protect a running pump from low flow, which could damage or destroy the pump. The software signals are implemented in a 2oo2 architecture, with the rest of the signals being 1oo1 like the simple inputs.

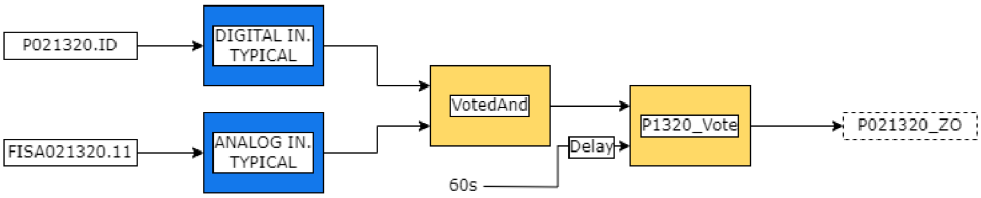


Fig. 3.14: Plant 73 - P1320 pump protection

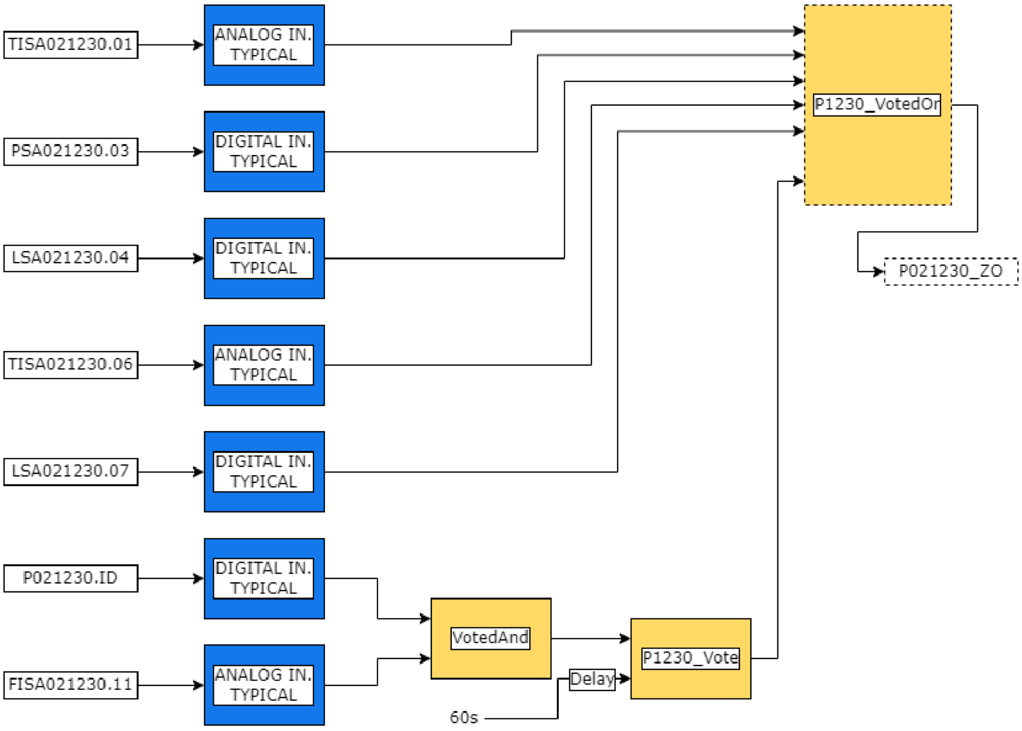


Fig. 3.15: Plant 73 - P1230 pump protection

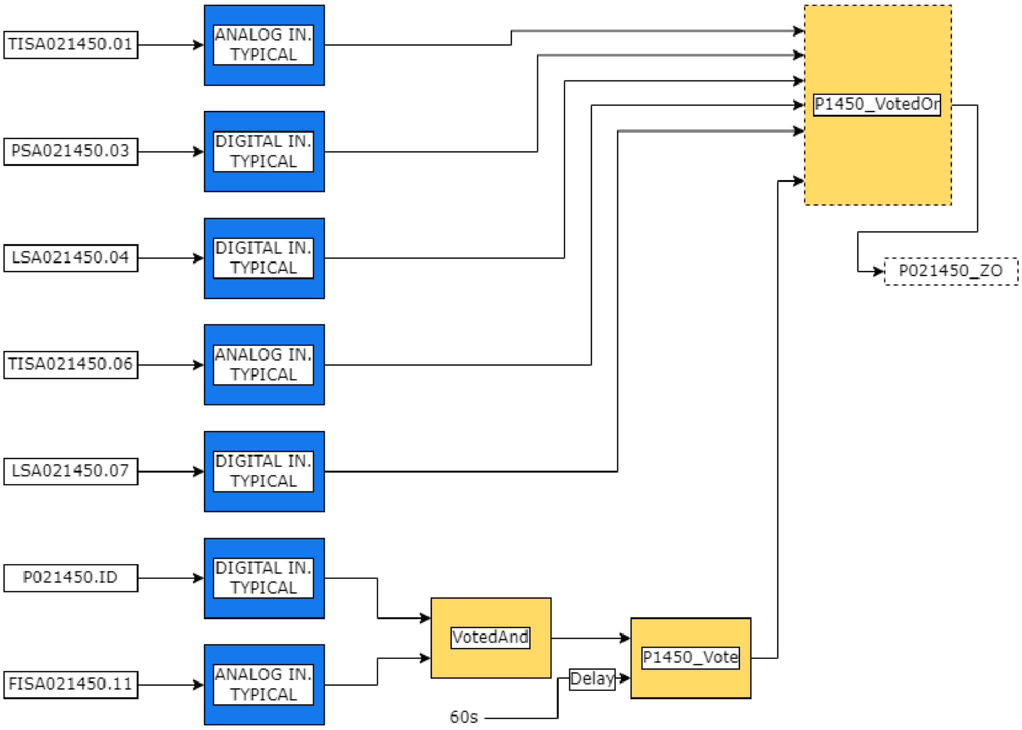


Fig. 3.16: Plant 73 - P1450 pump protection

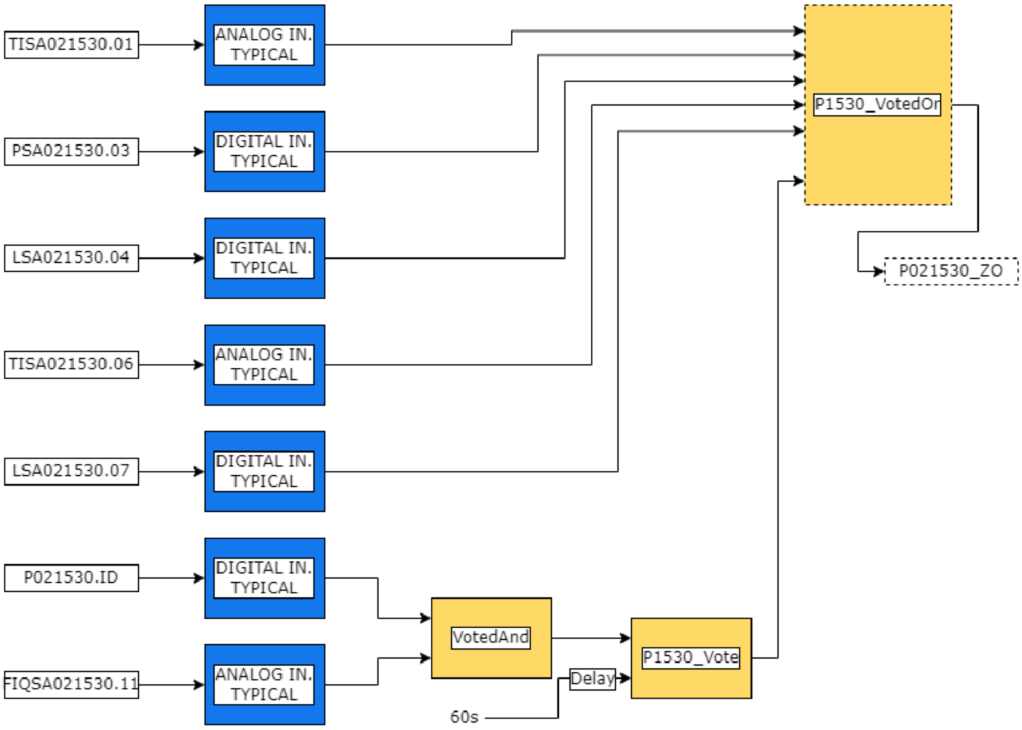


Fig. 3.17: Plant 73 - P1530 pump protection

Outputs

Outputs are divided into pages by the final element type. V outputs are safety valves and ZO outputs are the pumps.

Process phase switches are connected to a suppress command input of the digital output typical and inhibit the trip, with the exception of the 1210.V24ZO intake valve, which is tripped by activating the HS21000.01 signal.

The E-Stop signals have the biggest priority and always trip all the outputs, regardless of any process phases or forces being active.

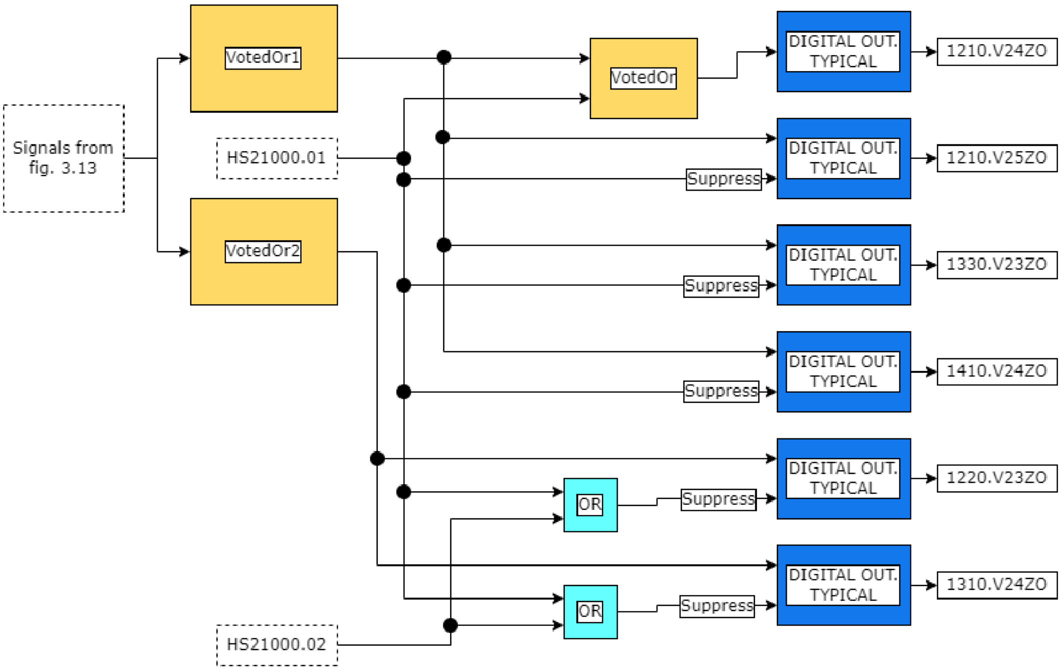


Fig. 3.18: Plant 73 - V outputs

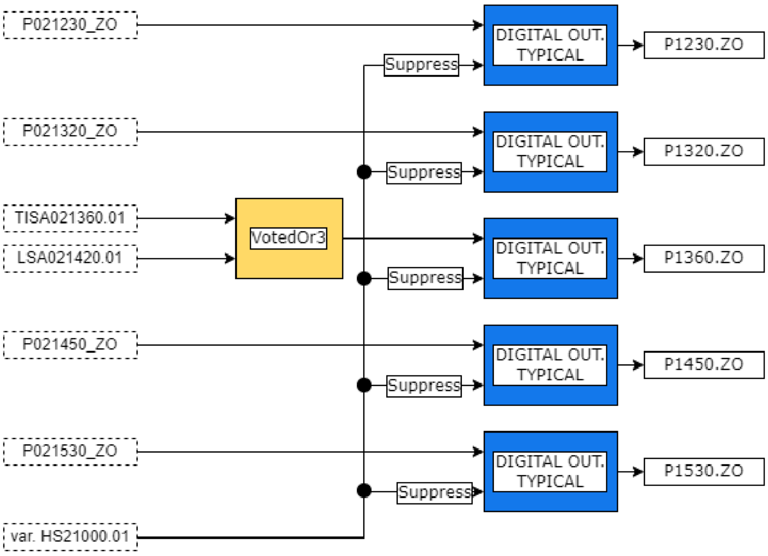
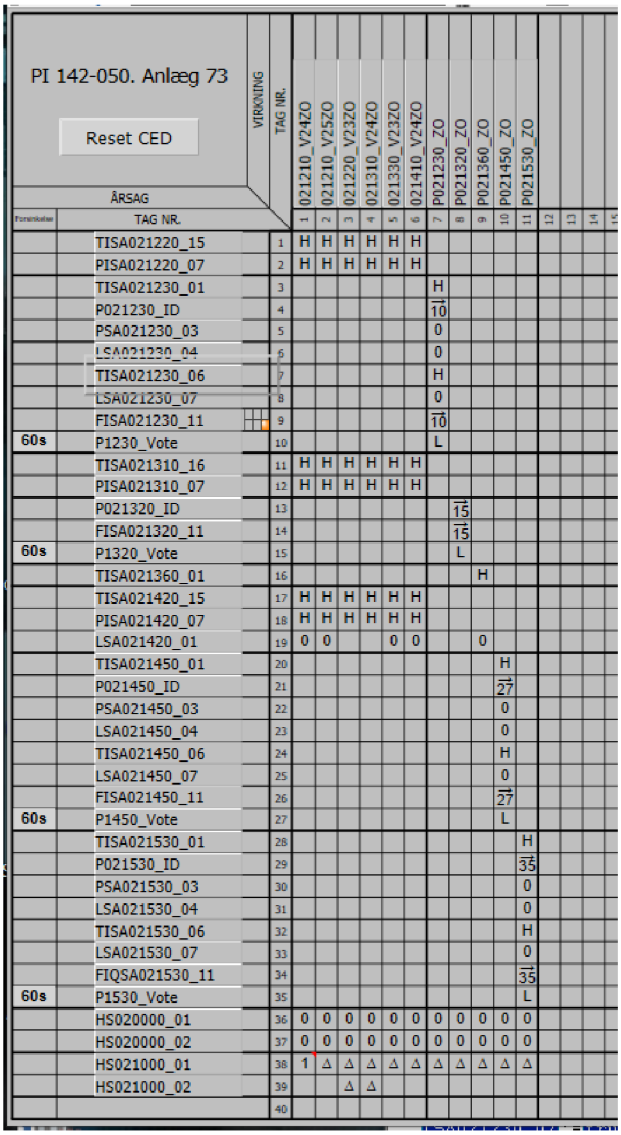


Fig. 3.19: Plant 73 - ZO outputs

3.4.3 Graphic displays

The displays were designed to look like the C&E matrices provided by the customer. The goal was to make it as easy as possible for the operators to monitor the entire system.

Every display contains a reset button to reset all logic in the respective unit, interactive elements to display individual faceplates for all inputs and outputs and a navigation element to quickly switch between units.



PI 142-050. Anlæg 73

	ÅRSAG	VIRKNING TAG NR.	021210_V24ZO	021210_V25ZO	021220_V23ZO	021310_V24ZO	021330_V23ZO	021410_V24ZO	P021230_ZO	P021320_ZO	P021360_ZO	P021450_ZO	P021530_ZO				
Forsinkelse	TAG NR.		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	TISA021220_15	1	H	H	H	H	H	H									
	PISA021220_07	2	H	H	H	H	H	H									
	TISA021230_01	3							H								
	P021230_ID	4							$\overrightarrow{10}$								
	PSA021230_03	5							0								
	LSA021230_04	6							0								
	TISA021230_06	7							H								
	LSA021230_07	8							0								
	FISA021230_11	9							$\overrightarrow{10}$								
60s	P1230_Vote	10							L								
	TISA021310_16	11	H	H	H	H	H	H									
	PISA021310_07	12	H	H	H	H	H	H									
	P021320_ID	13								$\overrightarrow{15}$							
	FISA021320_11	14								$\overrightarrow{15}$							
60s	P1320_Vote	15								L							
	TISA021360_01	16									H						
	TISA021420_15	17	H	H	H	H	H	H									
	PISA021420_07	18	H	H	H	H	H	H									
	LSA021420_01	19	0	0			0	0			0						
	TISA021450_01	20										H					
	P021450_ID	21										$\overrightarrow{27}$					
	PSA021450_03	22										0					
	LSA021450_04	23										0					
	TISA021450_06	24										H					
	LSA021450_07	25										0					
	FISA021450_11	26										$\overrightarrow{27}$					
60s	P1450_Vote	27										L					
	TISA021530_01	28											H				
	P021530_ID	29											$\overrightarrow{35}$				
	PSA021530_03	30											0				
	LSA021530_04	31											0				
	TISA021530_06	32											H				
	LSA021530_07	33											0				
	FIQSA021530_11	34											$\overrightarrow{35}$				
60s	P1530_Vote	35											L				
	HS020000_01	36	0	0	0	0	0	0	0	0	0	0	0				
	HS020000_02	37	0	0	0	0	0	0	0	0	0	0	0				
	HS021000_01	38	1	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ				
	HS021000_02	39			Δ	Δ											
		40															

Fig. 3.20: HMI display

3.5 Application design

3.5.1 SIL application

Finally the application implements only the top-level connections between units, "distributes" the E-Stop signal to all the units, implements the force enable functionality and handles the transfer of communication variable via IAC (Inter Application Communication) to the non-SIL application.

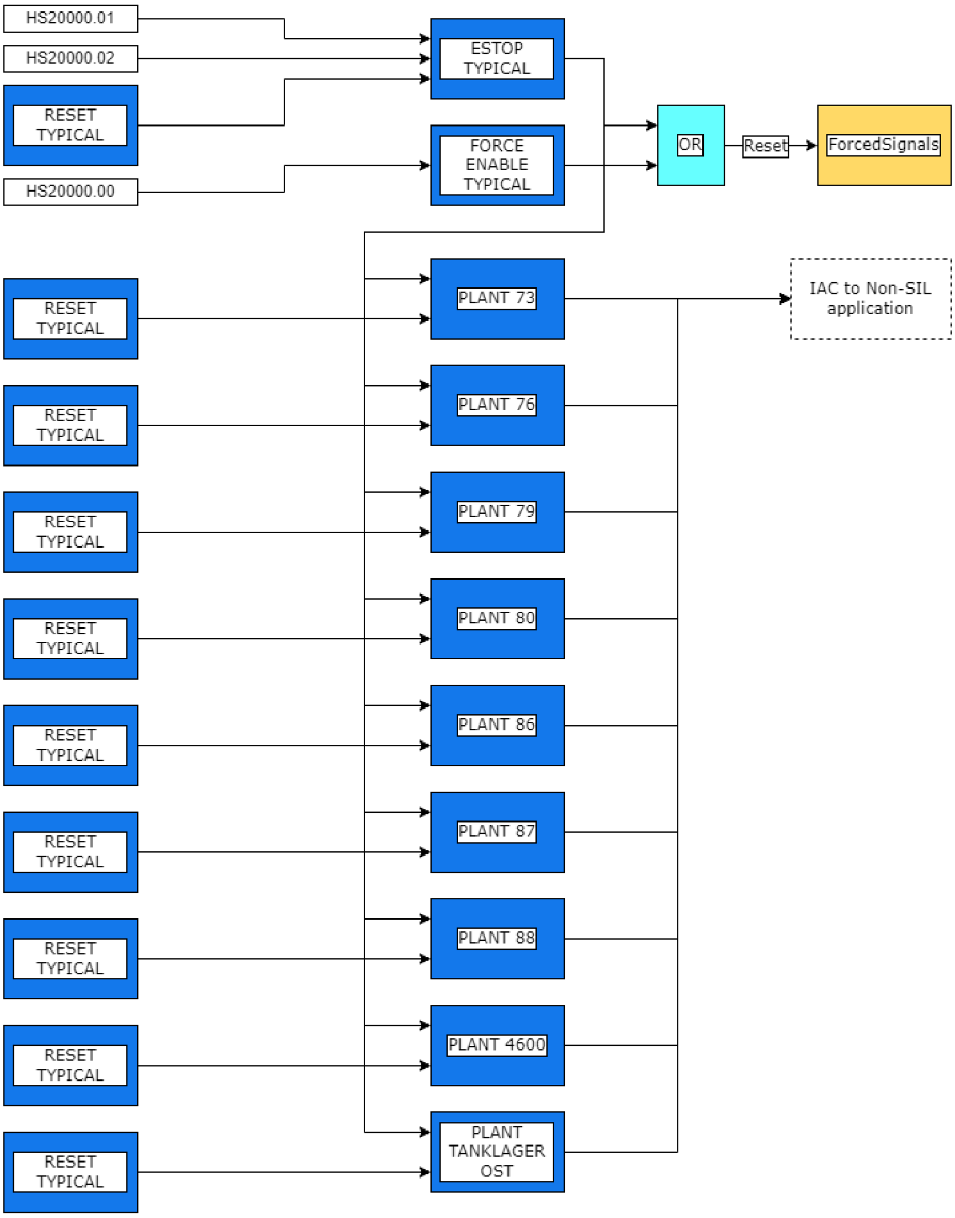
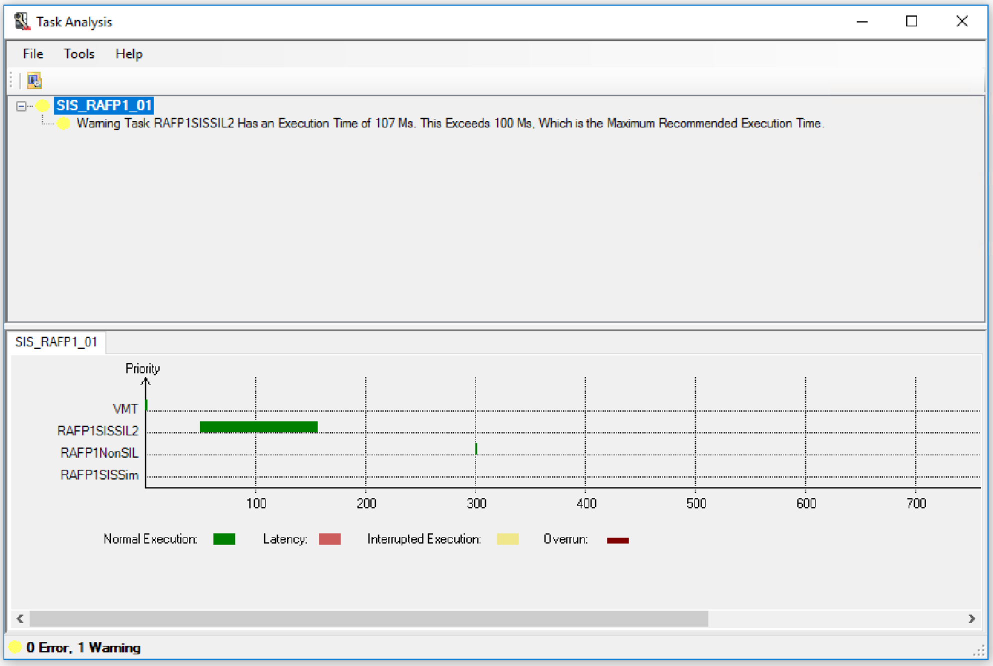


Fig. 3.21: SIL application

This approach was chosen to enable extensibility in the future, but it was later discovered that having a single task with many units exceeds the ABB 800xA system's maximum recommended task execution time of 100 ms by 7 ms and a warning is shown on download. This is normally not allowed in a SIL application, but due to only having a single task, therefore no risk of task collision it was agreed upon with the customer that the warning will be accepted during download.



SIS_RAFP1_01

Warning Task RAFP1SISSIL2 Has an Execution Time of 107 Ms. This Exceeds 100 Ms, Which is the Maximum Recommended Execution Time.

0 Error, 1 Warning

Fig. 3.22: Exceeded task execution time

3.5.2 Non-SIL application

This application handles the communication of values to the BPCS from SIS and it instantiates the communication typicals required for MasterBus 300 communication channel and connects them to the communication variables received from the SIL application. Apart from this, the application is used for internal alarms and diagnostics.

The communication variables need to be mapped to a single variable in order to be transferred. Boolean values are mapped in specific order to dint variables to be transferred as a single value, that can be decoded in the BPCS. The real values need to be sent separately as floating point numbers, due to data type limits of the communication channel.

4 Testing, Verification & Validation

The procedures used for all the activities outlined in this chapter are based on the requirements in clauses 7 and 12.5 of IEC 61511. The testing is done against the technical requirements specified in the SRCL and SFDS documents, according to a standard procedure outlined in the ABB FSM guidance document [5].

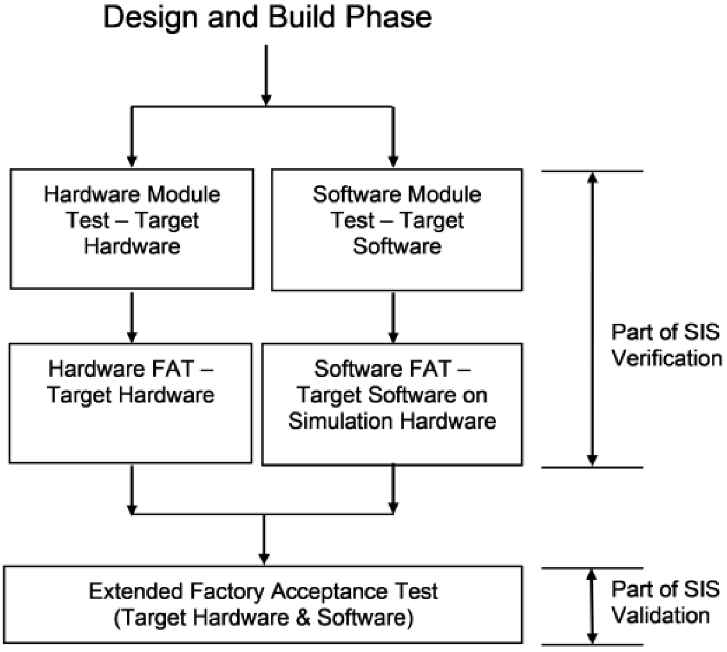

Design and Build Phase
Hardware Module Test – Target Hardware
Software Module Test – Target Software
Hardware FAT – Target Hardware
Software FAT – Target Software on Simulation Hardware
Part of SIS Verification
Extended Factory Acceptance Test (Target Hardware & Software)
Part of SIS Validation

Fig. 4.1: Test scenario [5]

In this test scenario, hardware and software are tested separately and the final integration is tested during the on-site validation.

The loop test will be performed by ABB Denmark to verify the correct functionality of the delivered components without the application program. During hardware FAT, the team from ABB Denmark will verify the IO allocation, perform an IO check and visually inspect the cabinet assembly.

Software was tested in a simulated environment and this chapter will be related only to the V&V activities relevant to the developed application program.

The final validation of the integrated software and hardware (EFAT) will be performed on-site during July 2024.

4.1 V-Model

To ensure that all requirements of the standard are met, the testing should be performed according to the V-Model defined by IEC 61511.

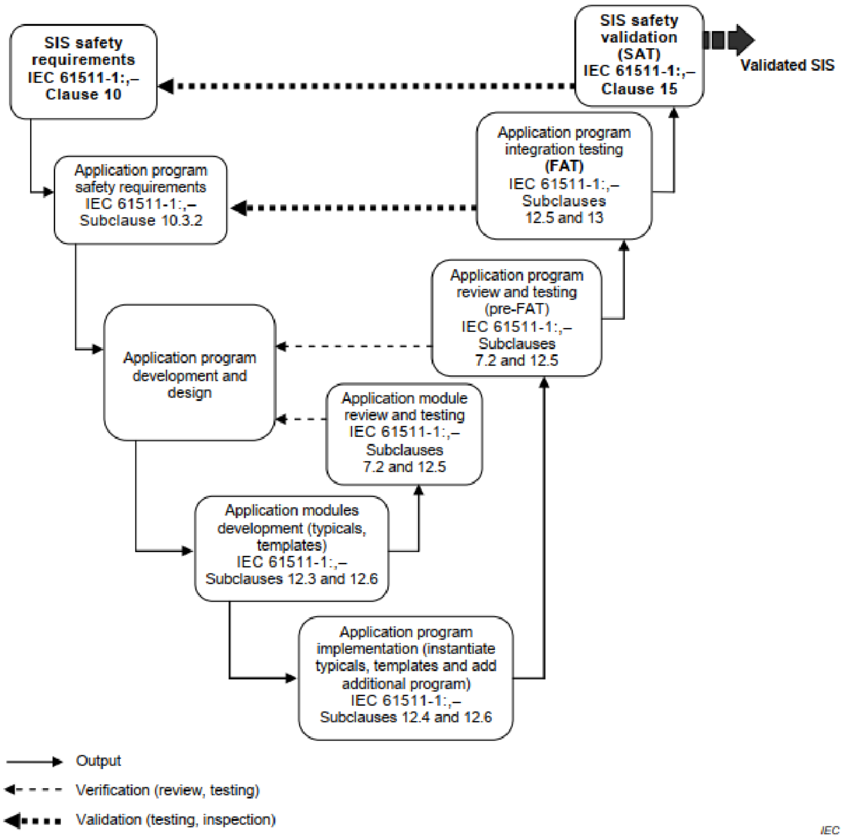

SIS safety requirements IEC 61511-1:,– Clause 10
SIS safety validation (SAT) IEC 61511-1:,– Clause 15
Validated SIS
Application program safety requirements IEC 61511-1:,– Subclause 10.3.2
Application program integration testing (FAT) IEC 61511-1:,– Subclauses 12.5 and 13
Application program development and design
Application program review and testing (pre-FAT) IEC 61511-1:,– Subclauses 7.2 and 12.5
Application module review and testing IEC 61511-1:,– Subclauses 7.2 and 12.5
Application modules development (typicals, templates) IEC 61511-1:,– Subclauses 12.3 and 12.6
Application program implementation (instantiate typicals, templates and add additional program) IEC 61511-1:,– Subclauses 12.4 and 12.6
Output
Verification (review, testing)
Validation (testing, inspection)
IEC

Fig. 4.2: Application program V-model [11]

This approach is practically impossible to achieve in a real world application as it would drastically increase both the cost of the project and the time necessary to complete the design and engineering.

In this specific project, the customer changed and/or updated the safety requirements specification multiple times during the development, which were handled as project queries and the complete testing process based on the V-Model was started only twice. During the first attempt, the testing process was stopped during the FAT in January 2024, due to the decision to implement more intended functionality and continuous amendments to the requirements. On ABB recommendation, the customer combined the raised project queries into a final work package, that the current versions of the application, documents and ultimately this thesis are based on.

The following sections are based on the testing activities performed during a re-test in May 2024.

4.2 Application program and document review

The goal of the review is to [5]:

- Verify whether the document or module correctly satisfies the specifications found in the reference documents
- Identify any deviation from standards, including issues that may affect maintainability of the Application Program
- Suggest improvement opportunities to the author

The review was performed by a senior competent person independent from the development team and concluded that no action needs to be taken and both the application program and documentation satisfy the specified requirements.

4.3 Module test

Module testing verifies that all the typicals developed for this project work as intended. In total, 17 tests were developed to test the behaviour during standard operation and all possible edge cases such as wire-breaks, overrides, limit values or delays [4].

Test cases are written specifically for each typical to test all possible combinations of parameters and situations. The tests are performed in a similar fashion to the following example:

Normal trip with 1oo1 voting - Digital input

1. Bring related inputs within healthy operation limits.
2. Reset vote object from related HMI via “Reset” button. Verify that no trip is present.
3. Initiate an alarm on the input via simulation application to the abnormal state.
4. Verify in Control Builder online that related output parameter “pDiffNormal” is active.
5. Verify trip output activation of vote object and output parameter.
6. Verify the logic, the faceplate and the graphic element indications & actions are as per SFDS/C&E.
7. Verify the alarm of input object is present in the Alarm list and have correct information.
8. Bring the input back to normal condition and acknowledge alarm of the related input object(s).
9. Reset vote object from related HMI via “Reset” button. Verify that no trip is present.

10. Verify in Control Builder online that reset of vote object occurs.

The tests were reviewed by the customer prior to the testing and performed by an independent safety engineer. One incorrect parameter was discovered in the analog input typical with 2 trip points and the issue was recorded in query list and rectified after the first module test. [4]

4.4 Internal Acceptance Test (IAT)

The IAT follows the same procedure as FAT, with the exception that it is done without the customer present and serves to identify any issues prior to testing the application program on FAT with the customer. Project settings, compiler switches, controller IP addresses and settings are verified against SFDS during IAT as well.

During the re-test, no issues were raised in the IAT phase, except for missing User Group definitions, which were recorded as accepted criteria for IAT and FAT. [4]

4.5 Factory Acceptance Test (FAT)

This phase is intended to verify that the developed application program satisfies all requirements of the SFDS and SRCL. The FAT is performed according to the flowchart in figure 4.3.

During the re-test it was noted that all tests including real hardware are postponed until the on-site test, which were recorded as accepted criteria for IAT and FAT.

On the first day of testing, the customer was not satisfied with the behaviour of the combined pump protection signals. This issue was resolved by submitting a project query and the input documentation will be amended by the customer to reflect the desired changes. It was decided that testing will continue as planned and the changes will be tested on-site as well.

Two more project queries regarding the missing zero delay indication on HMI displays and one incorrect signal trip point setting were noted and resolved after the test.

The test results were documented and test concluded as passed, with re-tests of project queries postponed to an on-site EFAT.

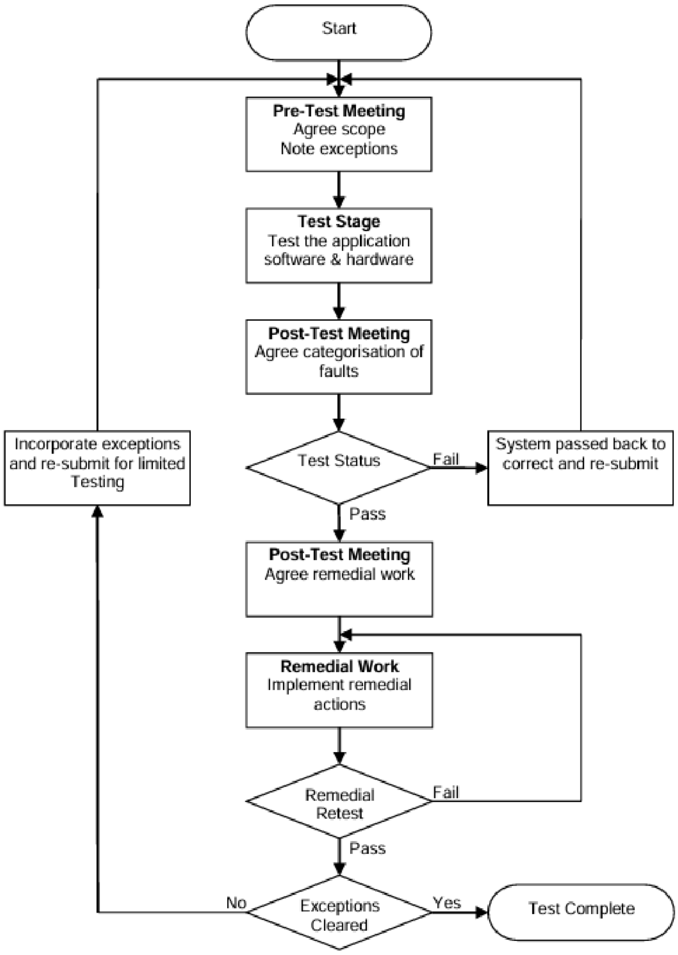


Fig. 4.3: Testing flowchart [4]

Conclusion

In this thesis, I have summarized the key aspects of IEC 61508 and IEC 61511 including the life-cycle approach, identification of safety targets, concept of safety integrity levels and the methods used to calculate the necessary risk reduction.

Next, I have given a brief overview of the scope and requirements of both above mentioned standards and shown their relationship, as well as highlighted their respective use-cases in different application sectors.

At the end of the first chapter, I have defined the scope of supply of an international contracting company in the role of a system integrator and included the specifics of ABB.

In the second chapter, I have summarized the safety requirements specification provided by the customer and highlighted the deviations from IEC 61511 and how the deviations are handled and resolved.

The chapter also deals with the developed documentation necessary to satisfy the requirements of all relevant clauses in IEC 61511.

In the next chapter, I have compiled a thorough, albeit simplified implementation of the application program and a quick look at the used hardware components.

During the application program development, I have been involved at every stage including typical, unit and application development. The typicals and units were developed collaboratively with or under the supervision of the software lead engineer, due to the requirements for experience and knowledge in terms of functional safety. This thesis only shows the latest version of the developed software solution, but over the course of the project multiple design variants were developed and considered.

In a similar manner, I have co-authored the deliverable project documentation in collaboration with the software lead engineer.

In the last chapter, I have documented the process of testing, verification & validation of the project. During the first attempt in January 2024, the testing process was halted due to a large number of project queries and addition of new intended functions. After the first attempt, the customer developed a final work package containing all the requested changes.

The second attempt took place in May 2024, which the developed application program passed with minor project queries, that were either resolved directly after or will be resolved prior to the on-site extended factory acceptance test in July 2024. All of the project queries were recorded as accepted criteria for the internal and factory acceptance tests and the project will proceed according to the execution plan and be fully validated and production started in summer 2024.

One of the biggest challenges, while writing this thesis were the numerous delays that pushed the FAT to one week before the submission deadline, which meant that parts of this thesis and the entire fourth chapter had to be rewritten.

Bibliography

[1] MONDŘÍK, Martin and MALYSA, Matěj, 2024. *Competency Assessment Form*. Internal document, PDF. RevA. Ostrava: ABB. [2023-08-25]

[2] MONDŘÍK, Martin and MALYSA, Matěj, 2024. *Safety Functional Design Specification*. Internal document, PDF. RevB. Ostrava: ABB. [2024-05-15]

[3] MONDŘÍK, Martin and MALYSA, Matěj, 2024. *Safety Requirements Checklist*. Internal document, PDF. RevB. Ostrava: ABB. [2024-05-15]

[4] MONDŘÍK, Martin and MALYSA, Matěj, 2024. *Test Specifications & Records*. Internal document, PDF. RevB. Ostrava: ABB. [2024-05-15]

[5] ABB PAEN FSM TA, 2024. *FSM Guidance*. Internal document, PDF. RevC. United Kingdom: ABB. [2024-01-01]

[6] ABB, 2024. *Functional safety management*. Online. Available at: `https://new.abb.com/oil-and-gas/systems-and-solutions/automation-and-safety/functional-safety-management`. [2024-05-15]

[7] ABB, 2023. *Function description. [HW Safety Functional Design Specification]*. Internal document, PDF. RevB. Ostrava: ABB. [2023-12-12]

[8] ABB, 2023. *ABB Ability™ System 800xA® hardware selector*. Online. Available at: `https://800xahardwareselector.com`. [2024-05-15]

[9] INTERNATIONAL ELECTROTECHNICAL COMMISSION, 2010. IEC 61508:2010 CMV Commented version, *Functional safety of electrical/electronic/programmable electronic safety-related systems - Parts 1 to 7*. Online. Edition 2.0. Geneva: IEC. Available at: `https://webstore.iec.ch/publication/22273`. [2024-05-15]

[10] INTERNATIONAL ELECTROTECHNICAL COMMISSION, 2017. IEC 61511-1:2016+AMD1:2017 CSV Consolidated version, *Functional safety - Safety instrumented systems for the process industry sector - Part 1: Framework, definitions, system, hardware and application programming requirements*. Online. Edition 2.1. Geneva: IEC. Available at: `https://webstore.iec.ch/publication/61289`. [2024-05-15]

[11] INTERNATIONAL ELECTROTECHNICAL COMMISSION, 2016. IEC 61511-2:2016 RLV Redline version, *Functional safety - Safety instrumented*

systems for the process industry sector - Part 2: Guidelines for the application of IEC 61511-1:2016. Online. Edition 2.0. Geneva: IEC. Available at: `https://webstore.iec.ch/publication/25521`. [2024-05-15]

[12] SMITH, David J. and SIMPSON, Kenneth G. L., 2020. *The safety critical systems handbook: a straightforward guide to functional safety: IEC 61508 (2010 edition), IEC 61511 (2016 edition) also related guidance on cyber security including machinery and other industrial sectors*. Fifth edition. Oxford: Butterworth-Heinemann / Elsevier.

Symbols and abbreviations

SIS	Safety Instrumented System
IEC	International Electrotechnical Commission
EFAT	Extended Factory Acceptance Test
SIL	Safety Integrity Level
$\mathbf{PFD_{avg}}$	Average probability of failure on demand
PFH	Probability of dangerous failure per hour
RRF	Risk Reduction Factor
ALARP	As Low As Reasonably Practicable
LOPA	Layer Of Protection Analysis
E/E/PE	Electrical/Electronic/Programmable Electronic
SIF	Safety Instrumented Function
FSMS	Functional Safety Management System
TÜV	Technischer Überwachungs-Verein
ABB	Asea Brown Boveri
SEC	Safety Execution Center
SRCL	Safety Requirements Checklist
SFDS	Safety Functional Design Specification
TS&R	Test Specification & Records
SRS	Safety Requirements Specification
C&E	Cause & Effect
IO	Input/Output
MTTR	Mean Time To Restoration
HMI	Human Machine Interface
BPCS	Basic Process Control System

MMS	Manufacturing Message Specification
DCS	Distributed Control System
CBM	Control Builder M
OOP	Object Oriented Programming
IAT	Internal Acceptance Test
FAT	Factory Acceptance Test
IP	Internet Protocol
FBD	Functional Block Diagram
1oo1	1 out of 1
E-Stop	Emergency stop
HS	Hand Switch
DO	Digital Output
IAC	Inter Application Communication
FSM	Functional Safety Management