Modicon Quantum

Quantum Safety PLC

Safety Reference Manual

10/2017

33003879.07

www.schneider-electric.com

Schneider Electric

The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither Schneider Electric nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

Table of Contents

Safety Information

Important Information

NOTICE

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, service, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

⚡ The addition of this symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.

⚠ This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ DANGER

DANGER indicates a hazardous situation which, if not avoided, **will result in** death or serious injury.

⚠ WARNING

WARNING indicates a hazardous situation which, if not avoided, **could result in** death or serious injury.

⚠ CAUTION

CAUTION indicates a hazardous situation which, if not avoided, **could result** in minor or moderate injury.

NOTICE

NOTICE is used to address practices not related to physical injury.

PLEASE NOTE

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

About the Book

At a Glance

Document Scope

This Safety Reference Manual describes the Quantum Safety PLC with special regard to how it meets the Safety requirements of the IEC 61508. It provides detailed information on how to install, run, and maintain the system correctly in order to protect human beings as well as to prevent damage to environment, equipment, and production.

This documentation is intended for qualified personnel familiar with Functional Safety and Unity Pro. Commissioning and operating the Quantum Safety PLC may only be performed by persons who are authorized to commission and operate systems in accordance with established Functional Safety standards.

Validity Note

This documentation is valid for Unity Pro from version 7.0.

Related Documents

You can download the Schneider Electric technical publications and other technical information from our website.

NOTE: All restrictions regarding electrical safety and external cabling and wiring must follow the documents in this table and the contents of this manual.

Title of Documentation	Reference Number
Modicon Quantum with Unity Ethernet Network Modules User Manual	33002479
Grounding and Electromagnetic Compatibility of PLC Systems User Manual	33002439
Modicon Quantum Hot Standby with Unity User Manual	35010533
Modicon Remote I/O Cable System Planning and Installation Guide	35014629
Premium, Atrium and Quantum using Unity Pro Communication services and architectures Reference manual	35006173
Quantum Instruction Sheets	33002365
Quantum TCPIP/IP Configuration User Manual	33002467
Quantum with Unity Pro Discrete and Analog I/O Reference Manual	35010516
Quantum with Unity Pro Hardware Reference Manual	35010529
Unity Pro Operating Modes Manual	33003101

Title of Documentation	Reference Number
Unity Pro OSLoader User Manual	35006156
Unity Pro Program Languages and Structure Reference Manual	35006144
Unity Pro Safety Block Library	33003873
Unity Pro XLS Operating Mode Manual Safety PLC Specifics	33003885
IEC 61131-2 Programmable controllers Part 2: Equipment requirements and tests, third edition 2007	–
IEC 61508 Functional safety of electrical/electronic/programmable electronic safety-related systems, edition 2.0	–
IEC 61511 Functional safety - safety instrumented systems for the process industry sector, edition 2016	–

You can download these technical publications and other technical information from our website at https://www.schneider-electric.com/en/download

Product Related Information

Schneider Electric assumes no responsibility for any errors that may appear in this documentation. Please contact us if you have any suggestions for improvements or amendments, or if you have found any errors in this publication.

No part of this documentation may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without written permission of Schneider Electric.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When controllers are used for applications with technical safety requirements, please follow the relevant instructions.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Use only Schneider Electric approved software.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Refer to IEC 61508, "Functional safety of electrical/electronic/programmable electronic safety-related systems".
- Completely understand the applications and environment defined by Safety Integrity Level (SIL) 3 within IEC 61508 Parts 1-7, edition 2.0.
- SIL requirements are based on the standards current at the time of certification.
- Do Not exceed SIL3 ratings in the application of this product.
- The terms identified in the list below as used in this document are applied only within the SIL3 rating.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Terms used in this document:

- certified
- failure (except as used in Special Message Statement of Consequence)
- fault
- non-interfering
- Quantum Safety PLC
- Quantum Safety CPU
- Safety analog inputs
- Safety analog module(s)
- Safety CPU
- Safety digital inputs
- Safety digital modules
- Safety digital outputs
- Safety FFB
- Safety firmware
- Safety I/O (module(s))
- Safety library
- Safety logic
- Safety memory area
- Safety modules
- Safety mode
- Safety outputs
- Safety PLC
- Safety power supply
- Safety programming
- Safety Quantum
- Safety-Related application(s)
- Safety remote I/O
- Safety variable

Chapter 1
General Information on the Quantum Safety PLC

Introduction

This chapter provides general information on the Quantum Safety PLC.

What Is in This Chapter?

This chapter contains the following sections:

Section 1.1
General Information

Introduction

This section provides information on the Quantum Safety PLC.

What Is in This Section?

This section contains the following topics:

IEC 61508 and Safety Integrity Level (SIL)

Introduction

The Quantum Safety PLC is a Safety-Related System certified according to IEC 61508 by TÜV Rheinland Group. It is based on the Quantum family of programmable logic controllers (PLCs). For programming, the Unity Pro XLS programming software of Schneider Electric must be used. Unity Pro XLS provides all the functionality of Unity Pro XL and is additionally able to program the Quantum Safety PLC. For further information on the differences between these software packages, see Differences between standard and Safety Quantum PLC *(see page 25)*.

IEC 61508 Description

The IEC 61508 is a technical standard concerning the Functional Safety of electrical, electronic or programmable electronic Safety-Related Systems.

A Safety-Related System is a system that is required to perform 1 or more specific functions to ensure risks are kept at an acceptable level. Such functions are defined as Safety Functions.

A system is defined functionally Safe if random, systematic, and common cause failures do not lead to malfunctioning of the system and do not result in injury or death of humans, spills to the environment, and loss of equipment and production.

Description of the Safety Integrity Level (SIL)

Safety Functions are executed to achieve and maintain the Safe state of a system. The IEC 61508 specifies 4 levels of Safety performance for a Safety Function. These are called Safety Integrity Levels (SIL), ranging from 1 (the lowest) to 4 (the highest). The Quantum Safety PLC is certified for use in SIL3 applications in which the de-energized state is the Safe state, for example in an emergency shutdown (ESD) system.

You can also use the Schneider Electric Safety products for creating a hot standby (HSBY) solution if you require high availability for a Safety-Related System.

Functional Safety Certification

Introduction

The Quantum Safety PLC is certified

- by TÜV Rheinland Group
- for use in applications up to and including SIL3 according to IEC 61508 and IEC 62061.

This certification verifies that the Quantum Safety PLC is compliant with the following standards:

- IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems, Part 1-7, edition 2.0
- IEC 61131: Programmable controllers
  - Part 2: Equipment requirements and tests, third edition 2007
- Boiler protection
  - US standards: NFPA 85 and NFPA 86
- EN 54 Fire detection and fire alarm systems
- EN 298 Automatic gas burner control systems for gas burners and gas burning appliances with or without fans
- IEC 62061: Safety of machinery
- EN ISO 13849: Safety of machinery

NOTE: Using a Quantum Safety PLC is a necessary but not sufficient precondition for the certification of a SIL3 application. A SIL3 application must also fulfill the requirements of the IEC 61508, IEC 61511, IEC 61131-2, and other application standards, see also *Requirements for Hardware and Programming, page 29*, *Exceptions and Requirements for Programming, page 72* and *Special Requirements for Application Standards, page 139*.

Classification of the Schneider Electric Products

The Quantum Safety PLC consists of Safety modules, which are allowed to perform Safety Functions. However, it also supports so-called non-interfering modules, thereby enabling you to add non-Safety parts to your SIL3 project.

Therefore, the Schneider Electric products must be distinguished into

- Safety modules and
- non-interfering modules.

In contrast to the Safety modules, non-interfering modules are not used to perform Safety Functions. They are certified as non-interfering modules for use in the Quantum Safety PLC. A fault in 1 of these modules does not influence the execution of the Safety Functions in a negative way.

Available Safety Products

Schneider Electric offers the following Safety modules certified for use in SIL3 applications. The Safety modules are listed with their corresponding PFD/PFH values for different proof test intervals (PTIs), see *Probabilities of Failure, page 20* and *Proof Test Interval, page 22*. The PFD/PFH are expressed as values that contributes to the overall PFD/PFH of the complete Safety loop (see *Safety Loop Description, page 20* and *Safety Loop Description, page 148*). The values are given for SIL3 applications.

The tables below list the Safety modules and their PFD/PFH values for **SIL3** applications:

Product Type	Product Reference	MTBF [h]	PTI = 1 year	
			PFD_G	PFH_G
Standalone Safety CPU	140 CPU 651 60S	600,000	1.527E-05	3.487E-09
Hot Standby Safety CPU	140 CPU 671 60S	600,000	1.527E-05	3.487E-09
Digital Input	140 SDI 953 00S	900,000	5.610E-07	1.218E-10
Digital Output	140 SDO 953 00S	1,000,000	7.156E-07	5.720E-11
Analog Input	140 SAI 940 00S	700,000	8.932E-07	7.770E-11
Power Supply (PS)	140 CPS 124 20	750,000	–	–
Power Supply (PS)	140 CPS 224 00	1,000,000	–	–

Product Type	Product Reference	PTI = 5 years	
		PFD_G	PFH_G
Standalone Safety CPU	140 CPU 651 60S	7.662E-05	3.507E-09
Hot Standby Safety CPU	140 CPU 671 60S	7.662E-05	3.507E-09
Digital Input	140 SDI 953 00S	2.806E-06	1.218E-10
Digital Output	140 SDO 953 00S	3.579E-06	5.727E-11
Analog Input	140 SAI 940 00S	4.467E-06	7.777E-11
Power Supply (PS)	140 CPS 124 20	–	–
Power Supply (PS)	140 CPS 224 00	–	–

Product Type	Product Reference	PTI = 10 years	
		PFD_G	PFH_G
Standalone Safety CPU	140 CPU 651 60S	1.540E-04	3.532E-09

Product Type	Product Reference	PTI = 10 years	
		PFD_G	PFH_G
Hot Standby Safety CPU	140 CPU 671 60S	1.540E-04	3.532E-09
Digital Input	140 SDI 953 00S	5.615E-06	1.219E-10
Digital Output	140 SDO 953 00S	7.160E-06	5.735E-11
Analog Input	140 SAI 940 00S	8.937E-06	7.785E-11
Power Supply (PS)	140 CPS 124 20	–	–
Power Supply (PS)	140 CPS 224 00	–	–

Product Type	Product Reference	PTI = 15 years	
		PFD_G	PFH_G
Standalone Safety CPU	140 CPU 651 60S	2.321E-04	3.557E-09
Hot Standby Safety CPU	140 CPU 671 60S	2.321E-04	3.557E-09
Digital Input	140 SDI 953 00S	8.426E-06	1.220E-10
Digital Output	140 SDO 953 00S	1.074E-05	5.744E-11
Analog Input	140 SAI 940 00S	1.341E-05	7.794E-11
Power Supply (PS)	140 CPS 124 20	–	–
Power Supply (PS)	140 CPS 224 00	–	–

Product Type	Product Reference	PTI = 20 years	
		PFD_G	PFH_G
Standalone Safety CPU	140 CPU 651 60S	3.109E-04	3.582E-09
Hot Standby Safety CPU	140 CPU 671 60S	3.109E-04	3.582E-09
Digital Input	140 SDI 953 00S	1.124E-05	1.221E-10
Digital Output	140 SDO 953 00S	1.433E-05	5.753E-11
Analog Input	140 SAI 940 00S	1.788E-05	7.803E-11
Power Supply (PS)	140 CPS 124 20	–	–
Power Supply (PS)	140 CPS 224 00	–	–

The Quantum Safety PLC is programmed with Unity Pro XLS.

CPU and IO detect the power supply errors, therefore the power supply does not contribute to the PFD/PFH values.

PCMCIA Memory Cards

The values in the Safety module tables above include the use of the following PCMCIA memory cards:

TSX MCPC 002M	TSX MRPC 768K
TSX MCPC 512K	TSX MRPC 001M
TSX MFPP 001M	TSX MRPC 01M7
TSX MFPP 002M	TSX MRPC 002M
TSX MFPP 004M	TSX MRPC 003M
TSX MFPP 512K	TSX MRPC 007M

Functional Safety Parameters

The Functional Safety parameters according to EN ISO 13849 are as follows:

- Performance Level for
  - SDI to SDO: PL d
  - SAI to SDO: PL d
- Category: 3

Available Non-Interfering Products

Schneider Electric offers the following non-interfering products:

Module Type	Module Reference
Remote I/O Head Adapter	140 CRP 932 00
Remote I/O Drop Adapter	140 CRA 932 00
Ethernet Module	140 NOE 771 11
Backplane 16 Slots	140 XBP 016 00
Backplane 10 Slots	140 XBP 010 00
Backplane 6 Slots	140 XBP 006 00
Digital Input	140 DDI 353 00
Digital Output	140 DDO 353 00
Analog Input	140 ACI 040 00
Analog Output	140 ACO 020 00
Terminal Strip	140 XTS 001 00
Terminal Strip	140 XTS 002 00
Remote I/O Optical Repeater	140 NRP 954 00

Module Type	Module Reference
Remote I/O Optical Repeater	140 NRP 954 01C
Time Stamping	140 ERT 854 20

⚠ **WARNING**

LOSS OF THE ABILITY TO PERFORM SAFETY FUNCTIONS

- Choose only Schneider Electric products certified for use in Safety-Related Systems in order to create a Safety-Related System.
- Use only Safety modules to perform Safety functions.
- Do not use inputs or outputs of non-interfering modules for Safety-Related outputs.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Unity Pro XLS offers modularization of the logic into sections. Schneider Electric recommends creating sections which are only used for non-Safety logic of the system. The data from non-interfering modules should be processed in these sections only, which makes the certification of your project much easier.

NOTE: To operate the Quantum Safety PLCs and to program and run your SIL3 project, you need the certified Safety version of the Quantum firmware. For details, see *Certified Products, page 22*.

Probabilities of Failure

For SIL3 applications, the IEC 61508 defines the following probabilities of failure on demand (PFD) and probabilities of failure per hour (PFH) depending on the mode of operation:

- PFD ≥ 10^{-4} to < 10^{-3} for low demand mode of operation
- PFH ≥ 10^{-8} to < 10^{-7} for high demand mode of operation

The Quantum Safety PLC is certified for use in low and high demand systems.

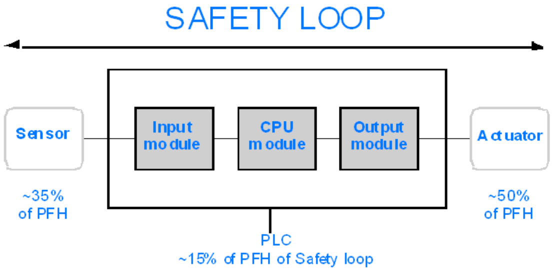
Safety Loop Description

The Safety loop to which the Quantum Safety PLC belongs consists of the following 3 parts:

- Sensors
- Quantum Safety PLC with Safety CPU and Safety I/O modules
- Actuators

Backplanes, a remote connection with CRA/CRP and Fiber Optic repeater modules do not destroy a Safety Loop. Backplanes, CRA/CRP and Fiber Optic repeater modules are part of a “black channel”. This means that the data exchanged by I/O and PLC cannot be corrupted without detection by the receiver.

The following figure shows a typical Safety loop:



For the calculation of the PFD/PFH values of an example system, a maximum of 15% is assumed for the PLC. For the PFD/PFH values of the Quantum Safety modules, see *Available Safety Products, page 17*.

NOTE: The programming tool Unity Pro XLS is not part of the Safety loop.

For detailed information on the IEC 61508 and its SIL policy, see also chapter *IEC 61508, page 143*.

Example Calculation

The following table gives 2 example calculations for PFD values within a SIL3 Safety loop with an assumed proof test interval of 10 years:

If the Safety loop contains ...	Then the PLC contributes to the Safety loop with ...	And sensors and actuators can use ...
● 1 digital input, ● 1 digital output, and ● a standalone CPU	5.610E-06 + 7.156E-06 + 9.979E-05 = 1.126E-04 => It corresponds to around 11.3% of the complete safety loop.	88.7%
● 2 sensors, ● 2 redundant analog inputs, ● 2 redundant digital outputs, and ● 2 Hot Standby CPUs	8.932E-06 + 7.156E-06 + 9.979E-05 = 1.159E-04 => It corresponds to around 11.6% of the complete safety loop. **Note:** All doubled modules contribute only once because the redundancy is only for high availability. Thus, only 1 module is active in the Safety loop.	88.4%

Safety Times Description

The Quantum Safety PLC has a minimum PLC cycle time of 20 ms, which is necessary for processing the signals from the I/O modules, executing the user logic, and setting the outputs. For calculating the maximum PLC reaction time, the maximum reaction time of the sensors and actuators you use must be known. Further, the maximum PLC reaction time depends on the process Safety time (PST) required for your process. You can find details of how to configure your PLC reaction time in *Process Safety Time, page 75*.

Proof Test Interval

The proof test is a periodic test performed to detect failures in a Safety-Related System so that, if necessary, the system can be restored to a like new condition or as close as practical to this condition. The time period between these tests is the proof test interval.

The proof test interval depends on the targeted Safety Integrity Level, the sensors, actuators and the PLC application. The Quantum is suitable for use in a SIL3 application and a proof test interval of 10 years. See Available Safety Products *(see page 17)* and Proof Test Procedure *(see page 30)*.

Certified Products

The Safety product versions are certified. Only certified versions are allowed for programming, commissioning, and operating the Quantum Safety PLC.

NOTE: Only Safety firmware can be loaded into the Quantum Safety PLC.

The Safety firmware is loaded with the OSLoader into the Quantum Safety PLC. Further information on how to load the firmware can be found in the Unity Pro OSLoader User Manual.

⚠ WARNING
Degrading the Safety Integrity Level
Only a CPU with Firmware Version 2.0 and above is suitable for SIL3.
A CPU with Firmware Version 1.0 is only suitable for SIL2 applications.
Failure to follow these instructions can result in death, serious injury, or equipment damage.

You can find the most recent information on the certified product versions on the TÜV Rheinland Group website *http://www.tuvasi.com/* under *Information* and further *List of Type Approved Programmable Electronic Systems*.

Special Operating Modes

Introduction

With regard to Functional Safety aspects, the following 2 operating modes of the Quantum Safety PLC are of special importance:

- the Safety Mode
- the Maintenance Mode

Safety Mode Description

The Safety Mode is the default mode of the Quantum Safety PLC, in which the Safety Functions are performed to control the process. It is a restricted mode in which modifications and maintenance activities are prohibited. You are only allowed to stop and start the PLC.

You can find a detailed description of the Safety Mode in *Safety Mode, page 90*.

Maintenance Mode Description

The Maintenance Mode of the Quantum Safety PLC is a temporary mode for debugging and maintaining your program. You are allowed to force values and to modify the program.

In the Maintenance Mode the (STOP or RUN), diagnostics are not available.

⚠ **WARNING**

LOSS OF ABILITY TO PERFORM SAFETY FUNCTIONS

In Maintenance Mode, all diagnostic functions are performed but their results are not fully evaluated. Once the Quantum Safety PLC exits Safety Mode and enters Maintenance Mode, you are fully responsible for ensuring the Safe state of your system.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

You can find a detailed description of the Maintenance Mode in *Maintenance Mode, page 92*.

Diagnostics

Introduction

The Quantum Safety PLC provides additional internal diagnostics and system testing, increasing the diagnostic coverage (DC).

Survey of the Diagnostics

The internal architecture of the Quantum Safety CPU

- provides 2 shutdown paths and
- allows double code generation and execution to detect
  - systematic faults in the code generation and execution and
  - random faults in the CPU and the RAM.

  The double code execution is controlled by 2 different processors integrated into the CPU.

For further details, see *Standalone Safety CPU, page 33*.

The internal architecture of the Quantum Safety I/O modules

- provides redundancy,
- detects systematic faults in the code execution, and
- random faults in the I/O modules.

The communication between the CPU and the I/O is designed as a black channel. The protocol checks or manages detected errors such as detected transmission errors, omissions, insertions, wrong order, delays, incorrect addresses, and masquerade bits, and retransmissions. Therefore, the non-interfering modules such as backplanes, Fiber Optic repeaters (140 NRP 954 00, 140 NRP 954 01C), remote I/O adapters 140 CRP 932 00 and 140 CRA 932 00 can be used inside the safety loop without impact on the PFD and PFH evaluations.

For further details, see *General Information on the Safety I/O Modules, page 39*.

Difference Between Standard Quantum PLC and Quantum Safety PLC

Differences Between Standard and Safety PLC

To meet the requirements of the IEC 61508 standard, the Quantum Safety PLC differs from the standard Quantum PLC.

The following table lists the main differences between a standard Quantum and a Safety Quantum PLC:

Feature	Standard Quantum PLC	Quantum Safety PLC
CPU Program Execution	executed on application processor or Intel	executed on application processor and Intel
Configuration	● backplane ● local rack ● remote I/O ● all power supplies ● backplane expanders ● distributed I/O ● fieldbus I/O	● backplane ● local rack ● remote I/O ● dedicated power supply
Firmware	regular firmware	Safety firmware
Software	● Unity Pro XLS ● Unity Pro XL ● Unity Pro L	● Unity Pro XLS
User Logic	● FBD ● LD ● IL ● ST ● SFC	● FBD ● LD
Data Type	● EDT ● DDT	● EDT ● only simple arrays
Mode	–	● Maintenance Mode ● Safety Mode
Restart Behavior	● no restart ● cold start ● warm start	● no restart ● cold start

Differences Between Standard and Safety PLC OS

To meet the requirements of the IEC 61508 standard, the operating system (OS) of the Quantum Safety PLC differs from that of the standard Quantum PLC.

The following table lists the main differences between a standard Quantum PLC OS and a Safety Quantum PLC OS:

Feature	Standard Quantum PLC OS	Quantum Safety PLC OS
Warm Start	yes	no
Safety Mode	no	yes
Minimal Time Duration for MAST Execution in Cyclic Mode	3 ms	20 ms
Forcing Safety Mode by Locking the Key	no	yes
Display of Mode Indicating Characters on LCD	no	yes
Memory Check	no	yes
Password	no	yes
Safety Analog Input	no	yes
Safety Digital Input	no	yes
Safety Digital Output	no	yes
Meaning of SW12, SW13	no	Safety mode
MSTR Blocks	yes	no
Global Data Subscribing (Ethernet)	everywhere	only in unrestricted area
I/O Scanner Read (Ethernet)	everywhere	only in unrestricted area
Global Input and Specific Input (Modbus Plus)	everywhere	only in unrestricted area
Unrestricted Area for %M and %MW	no	yes

Notes

The Quantum Safety PLCs only perform cold start. Thus, the application is reinitialized at each start.

The Quantum Safety PLC can run in cyclic or periodic mode. Thus, there is no difference in its behavior compared to the standard Quantum PLC. For details on cyclic and periodic execution, see the chapter "Application Program Structure" in the *Unity Pro Program Languages and Structure Reference Manual*.

Memory

The memories of the Quantum Safety CPUs are each divided into a Safety and an unrestricted part. The Safety memory area is write protected and used for processing Safety-Related data. The unrestricted memory area is not write protected and used if it is necessary to get access to the Safety Functions. Its values cannot be used directly but by using specific function blocks, see *Memory Area Description, page 104*.

In slot A, PCMCIA memory cards can be used in a Quantum Safety CPU in the same way as they can be used in a standard Quantum CPU. These cards can be standard type, application and file-type or data and file-type memory cards. For details on this topic, see the chapter "High End CPU" in the *Quantum with Unity Pro Hardware Reference Manual*.

In contrast, slot B for data and file-type memory cards is not allowed to be used because this data storage is not available for SIL3 projects.

⚠ WARNING

LOSS OF ABILITY TO PERFORM SAFETY FUNCTIONS

Do not use slot B. Data stored on a memory card in slot B is not processed in SIL3 projects.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Hot Standby

In addition to the standard Quantum Hot Standby functions, you can also use the Quantum Safety PLCs for Safety-Related Hot Standby systems in order to achieve high availability for the CPU in a Safety-Related System. To control the ability of the standby PLC to take over from the primary, you can use an elementary function block (EFB) to program an automatic swap between primary and standby PLC. For further information on this topic, see also *Hot Standby Safety CPU Specifics, page 35*.

Redundant I/O

To achieve high availability for the I/O, you can also use the Safety I/Os in a redundant manner. For further information on this topic, see also *Configuration Examples for the Quantum Safety PLC, page 62*.

Training

Introduction

As stated in the IEC 61508, Part 1, App. B, all persons involved in a Safety Lifecycle activity should have the appropriate training, technical knowledge, experience, and qualifications relevant to the specific duties they have to perform. This should be assessed in relation to each particular application.

NOTE: Make sure you possess all information and skills required to install, run, and maintain Safety-Related Systems correctly.

Training Contents

In addition to the usual training courses concerning the use of the company’s products, Schneider Electric offers you training courses covering the topics of its IEC 61508 compliant Safety-Related System.

Section 1.2
Safety Requirements

Requirements for Hardware and Programming

Introduction

You must fulfill the following Safety requirements when using the Quantum Safety PLC.

Hardware Requirements

- For a SIL3 project, you must use 1 of the 2 following Quantum Safety CPUs:
  - 140 CPU 651 60S for stand-alone systems
  - 140 CPU 671 60S for systems requiring high availability
- Only Quantum Safety modules are allowed to perform Safety Functions. Non-interfering modules can be part of the Safety PLC because they do not interfere with the Safety modules by their own means. However, they are not allowed to execute Safety Functions. They can only be used to process non-Safety signals except the backplanes and remote IO adapters, which are considered as part of a black channel.
- The Safe state of the outputs is the de-energized state.
- You must follow the specified operating conditions regarding EMC, mechanical, and climatic influences. For details, see the chapter "System Specifications" in the *Quantum with Unity Pro Hardware Reference Manual*.

NOTE: Backplane expanders and distributed I/Os are not allowed in the Quantum Safety PLC configuration.

NOTE: All Safety and non-interfering modules fulfill the requirements of the IEC 61131-2.

Programming Requirements

- For programming a SIL3 project, you must use the certified Quantum Safety firmware and the Safety programming software Unity Pro XLS.
- You must make sure that your SIL3 project is configured and programmed correctly according to the rules of the IEC 61508 as well as to the rules described in this Safety Reference Manual.
- For the complete life-cycle of the project development, you must follow the requirements of the IEC 61511 for installation, commissioning, and validation.
- The logic can be tested in simulation mode but the full test of the Safety Functions must be performed with the runtime system and the complete installation.

⚠ WARNING

RISK OF PROJECT ERRORS

Check that your project is correct according to your specification by performing tests on the runtime system.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Proof Test Procedure

The user must perform the proof test procedure periodically (see IEC61508-4, 3.8.5). The maximum time between 2 proof test is the proof test interval.

For the safety PLC itself, the proof test consists of:

- A power cycle
- Checks that all modules restart without a detected diagnostic error

In addition, a complete commissioning of the safety application has to be performed. The complete procedure must include the necessary tests of cabling, sensors and actuators, depending on the full application analysis.

Chapter 2
Hardware and Configuration

Introduction

This chapter provides information concerning hardware and configuration of the Schneider Electric products that can be used for SIL3 applications.

What Is in This Chapter?

This chapter contains the following sections:

Section 2.1
Safety CPU

Introduction

The following section introduces the internal architecture of the Quantum Safety CPUs, distinguished according to their use in standalone and Hot Standby solutions.

What Is in This Section?

This section contains the following topics:

Standalone Safety CPU

Introduction

For use in standalone SIL3 solutions, the **140 CPU 651 60S** Quantum Safety CPU is certified.

The safety CPU includes a PCMCIA memory card, but its use and presence is not mandatory.

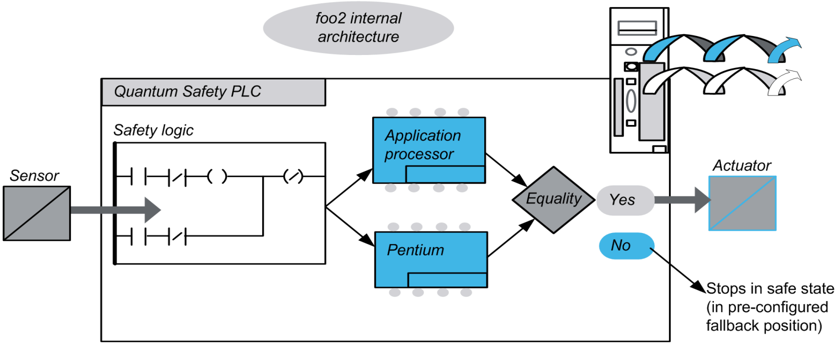
Description of the Internal CPU Architecture

The Quantum Safety CPU contains 2 different processors, an Intel Pentium and an application processor. Each one executes the Safety logic in its own memory area and both compare the results of the execution at the end of each cycle.

Two CPUs are available:

- 140 CPU 651 60S (Standalone Safety CPU)
- 140 CPU 671 60S (Hot Standby Safety CPU)

The following figure shows the internal architecture of the Quantum Safety CPU:



Benefits of the Double Code Generation and Execution

The 2 processors inside the Quantum Safety PLC allow double code generation and execution.

This diversity provides the following advantages in error detection:

- 2 executable codes are generated independently. The diversity of compilers allows the detection of systematic error in the code generation.
- The 2 generated codes are executed by 2 different processors. Thus, the CPU is able to detect both systematic errors in the code execution and random errors in the PLC.
- 2 independent memory areas are used for the 2 processors. Thus, the CPUs are able to detect random errors in the RAM and a full RAM test is not necessary at every scan.

Description of the Watchdog

A hardware and a firmware watchdog check the PLC activity and the time needed to execute the user logic.

NOTE: You must configure the software watchdog (maximum PLC cycle time) to be consistent with the application execution time, the filtering of the I/O communication error, and the process Safety time (PST) targeted.

Description of the Memory Check

Static memory areas, including the Flash memory, PCMCIA memory card (if any) and the RAM, are checked using the cyclic redundancy check (CRC) and the double code execution. Dynamic areas are protected by the double code execution and a periodic memory test. At cold start, these tests are re-initialized and fully performed before the CPU goes into Stop or Run mode.

Hot Standby Safety CPU Specifics

Introduction

The 140 CPU 671 60S Quantum Safety CPU module is certified for use in Hot Standby SIL3 solutions compliant with the 61508 IEC standard. For more details about to the safety certifications, refer to the *Modicon Quantum Safety PLC Safety Reference Manual*.

In the Standalone Safety CPU, the Ethernet port is used to communicate with other devices using a normal Ethernet cable.

In the Hot Standby Safety CPU, the connection used to exchange data between the Primary CPU and the Standby CPU controller is a fiber optic link. Because the fiber optic link is not part of the Safety loop, the PFD and PFH values of the Hot Standby CPU are the same as those of the Standalone CPU.

Each Safety CPU can include a PCMCIA memory card, but its use and presence is not mandatory.

NOTE: This CPU cannot be used in a Quantum Ethernet I/O Hot Standby system.

Description of a Safety Hot Standby Configuration

The Hot Standby configuration contains two identical local racks and at least one remote I/O drop because I/Os cannot be placed in the local rack of a Safety Hot Standby configuration.

Besides a power supply module (there must be at least one 140 CPS 124 20 or one 140 CPS 22 400), each local rack must contain:

- 140 CPU 671 60S module
- 140 CRP 932 00 module

Besides a power supply, I/O modules (including at least one 140 CPS 124 20 or one 140 CPS 22 400), the remote drop(s) must include a 140 CRA 932 00 module.

⚠ **CAUTION**

UNINTENDED EQUIPMENT OPERATION

Use only high availability RIO modules with dual cabling in a Safety-Related System.

Failure to follow these instructions can result in injury or equipment damage.

Description of the Operating Modes

- **Safety Mode:** This is the default mode. It is a restricted mode in which modifications and maintenance activities are prohibited.
- **Maintenance Mode:** This is a temporary mode for modifying the project, debugging and maintaining the application program.

State Compatibility with Safe and Maintenance Modes

A Quantum Hot Standby system has two states:

- **Redundant (1 CPU is Primary, 1 is Standby)**
  The Standby CPU controller mode follows the Primary CPU controller mode. For example, if you switch the Primary CPU controller from Safety to Maintenance mode, the Standby CPU controller switches from Safety to Maintenance mode at the start of the next cycle.
- **Non-redundant (at least 1 CPU Offline)**
  The two controllers are independent, one can be in Safety mode and the other one in Maintenance mode. For example, the Run Primary controller can be in Safety mode while the Stop Offline controller is in the Maintenance mode.

Impact of the PLC Switchover on the Process Safety Time

If the Primary CPU detects an internal or external problem, it stops exchanging data with the Standby CPU and stops processing the I/O. As soon as the Standby CPU detects that there are no longer exchanges with the Primary CPU, it takes over the role of the Primary CPU, executing the user logic and processing the I/O. Therefore, the output modules must filter the lack of exchange with the Primary CPU to avoid glitches when a Switchover occurs. This is achieved by configuring the output module time-out. As a result, the PLC reaction time is greater than the time-out configured in the output module, thereby influencing the process Safety time.

NOTE: The behavior of the Hot Standby Safety CPU is equivalent to a Standalone Safety CPU.

In case of a detected error, the Safety PLC enters:

- Halt state when running in the Maintenance Mode
- Error state when running in the Safety Mode

Availability of the Hot Standby Functions

In addition to the standard Hot Standby functions, you can use an EFB to program an automatic Switchover between Primary CPU and Standby CPU to verify the ability of the Standby CPU to take over from the Primary CPU. That means that the Standby CPU periodically becomes the Primary CPU and the Primary CPU becomes the Standby CPU.

It is recommended to avoid using the USB link during a Switchover.

The following table lists the available Hot Standby functions in Maintenance and Safety modes:

Function	Maintenance Mode	Safety Mode
Hot Standby	yes	yes
Switchover	yes	yes
EFB Swap	no	yes
Keypad	yes	yes
Application mismatch	yes	no
OS Upgrade	yes, if Standby CPU is in Stop Offline	no
Application Transfer	yes	no

NOTE: Applying the power simultaneously to Primary CPU and Standby CPU is allowed, but we recommend to do it sequentially.

Section 2.2
Safety I/O Modules

Introduction

This section deals with the 3 Safety I/O modules developed for use in the Quantum Safety PLC. The features that the 3 modules share are described in general, whereas their distinctions are explained separately.

What Is in This Section?

This section contains the following topics:

General Information on the Safety I/O Modules

Introduction

The following 3 Quantum Safety I/O modules are certified for use in safety applications:

- 140 SAI 940 00S (Analog Input)
- 140 SDI 953 00S (Digital Input)
- 140 SDO 953 00S (Digital Output)

The 3 Safety I/O modules allow you to connect the Safety PLC to the sensors and actuators, which are part of the Safety loop. All of them are composed of 2 micro controller systems running the same program, sharing the same information and checking each other periodically. You can install these I/O modules in the local backplane or in remote I/O drops.

Description of the CPU-I/O Communication

In general, the Quantum Safety CPU masters all backplane exchanges whereas the other modules are slaves. Between Safety CPU and Safety I/Os, data are exchanged through a dual port RAM, located in the I/O module.

For the communication between CPU and remote I/Os (RIOs), you must use the following 2 non-interfering modules:

- 140 CRP 932 00 (RIO head adapter), located in the local rack
- 140 CRA 932 00 (RIO drop adapter), located in the RIO drop

Optionally, you can use Fiber Optic repeater modules (140 NRP 954 00, 140 NRP 954 01C). Those modules enhance remote I/O network noise immunity and increase cable distance while maintaining the full dynamic range of the network and the safety integrity level.

The communication protocol between the Safety I/O and CPU secures their exchanges. It allows both to check the correctness of received data, and detect any failure of the transmitter or during the transmission. Thus, a safety loop may include any non-interfering RIO adapters and backplane. For details on this topic, see Safety I/O Modules Diagnostics *(see page 43)*.

The Safety I/O modules provide features for line monitoring, see Safety I/O Modules Diagnostics *(see page 43)* and the Quantum with Unity Pro Discrete and Analog I/O Reference Manual.

NOTE: Use the red labels provided with the Quantum Safety /IO modules to clearly indicate the Safety modules.

Safety I/O Modules in High Availability Configurations

Introduction

The Quantum Safety I/O modules can be used in a redundant way to increase the availability. However, using redundant Safety modules does not increase Safety. Schneider Electric provides function blocks to supervise the state for a configuration with 2 redundant modules.

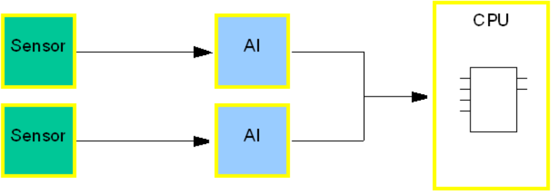
The health of the modules is available by system words, which you can evaluate and make available to the operator and maintenance personnel in order to inform them in case a module is inoperable and must be exchanged. Each bit in the word represents the health of one channel. For further details, see the chapter "Quantum Safety I/O Modules" in the *Quantum with Unity Pro Discrete and Analog I/O Reference Manual*. The system is still running in a SIL3 configuration and the only time limit for the exchange of the module is the proof test interval.

The modules can be placed in the same drop. However, Schneider Electric recommends using different drops to avoid problems in a single drop (remote adapter or power supply outage), see also chapter *Configuration Examples for the Quantum Safety PLC, page 62*.

High Availability Analog Input Modules

2 sensors must be used for high availability Safety analog inputs and each must be connected to a different input point. The 2 input points must be located on different input modules.

The following figure illustrates the redundant analog input configuration:

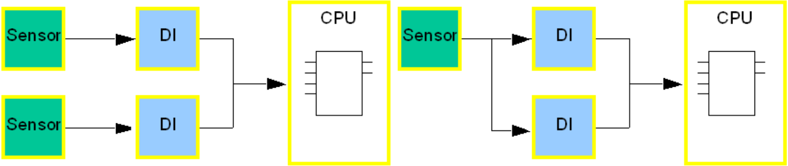


The function block `S_AISIL2`, see also *Functions/Function Blocks for SIL3 Applications, page 82*, can be used for selecting the data from the 2 redundant analog inputs and to supervise the state of the inputs.

High Availability Digital Input Modules

The redundant Safety digital inputs can be connected to either 1 or 2 sensors. The 2 input points must be located on different input modules. In case you use 1 sensor, the modules share the same process power supply. Using the information on the modules (input characteristics on short circuit, open wire, zero and one level, voltage and current) specified in the Quantum with Unity Pro Discrete and Analog I/O Reference Manual, you must define the wiring to fulfill these characteristics.

The following figure illustrates the redundant digital input configurations:

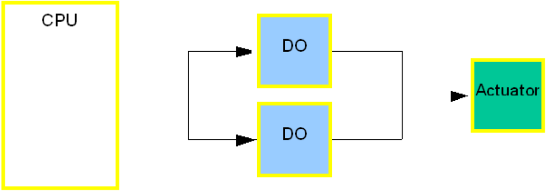


The function block S_DISIL2, see also *Functions/Function Blocks for SIL3 Applications, page 82*, can be used for selecting the data from the 2 redundant digital inputs and to supervise the state of the inputs.

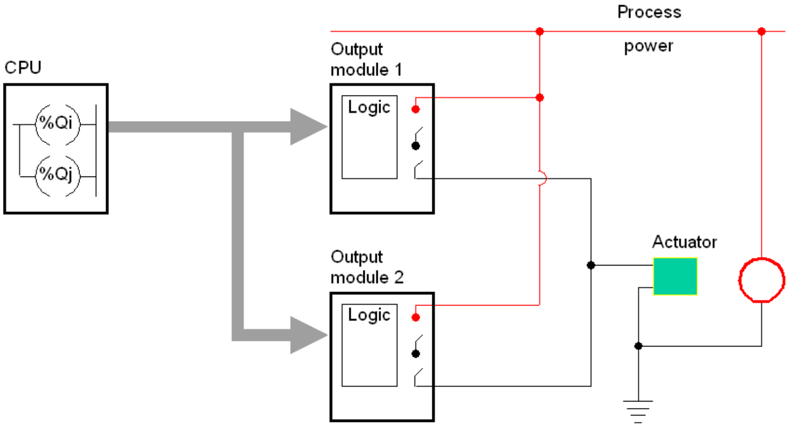
High Availability Digital Output Modules

For high availability digital outputs, the 2 outputs must be on separate modules, wired in parallel and connected to 1 actuator.

The following figure illustrates the redundant digital output configuration:



The following figure shows the electrical scheme for this configuration:



A function block is not necessary because the same signal from the CPU is connected to both outputs.

Safety I/O Modules Diagnostics

Description of the I/O Diagnostics

The following table lists the field diagnostics of the Safety I/O modules:

Diagnostics	Analog Input	Digital Input	Digital Output
Out of Range	yes	–	–
Broken Wire	yes	yes	–
Field Power	–	yes	yes
Overload	–	–	yes

NOTE: Short circuit of the wiring is not detected for the input modules. It is your responsibility to make sure that the modules are wired correctly, see the Quantum with Unity Pro Discrete and Analog I/O Reference Manual.

In addition, the Quantum Safety PLC provides diagnostics of the communication between Safety CPU and Safety I/O modules, for instance a CRC. Thus, it is not only checked that the data received are the data sent but also that the data are updated. To handle disturbances, for example by EMC effects, which may temporarily corrupt your data, you can configure a maximum accepted consecutive CRC error for each module (ranging from 1 to 3). For a detailed procedure, see the chapter “Configuring I/O Modules for Safety Projects” in the *Unity Pro XLS Operating Mode Manual Safety PLC Specifics*.

Diagnostics at Power Up

At power up, the I/O modules perform an extended self-test during about 30 seconds. If these tests are unsuccessful, the modules are not considered to be healthy and do not start. The inputs and outputs are then set to 0.

If the 24 VDC external power supply is not connected to the digital input or digital output modules, the power up self-tests do not take place and the modules does not start.

Runtime Diagnostics

During runtime, the I/O modules perform self-tests. The input modules verify that they are able to read data from the sensors over the complete range. The output modules perform pulse tests on their switches with a duration lower than 1 ms.

Description of the General Over Voltage Diagnostics

Because the electronics may not work up to the theoretical maximum output voltage of the power supplies, the I/O modules must supervise the backplane power supply voltage.

The following table describes the supervision of the power supply:

The power supply of ...	Is monitored by ...
the backplane, which has a theoretical maximum output voltage of 18.5 V,	2 over voltage supervisors, that is 1 for each micro processor system. Each supervisor is able to handle a possible over voltage by opening its power switch and triggering its reset block, which manages transitions between the states of power on and power off and resets both processors when active.
the field side, which is generated by DC-to-DC converters,	2 over and under voltage supervisors, that is 1 for each micro processor system. If the 2 isolated DC-to-DC converters generating the power supply to the field side electronics experience a fault, the supervisors signal this condition to its particular processor through an isolator.
the process, which is one of the PELV type with a maximum output voltage of 60 V,	2 over and under voltage supervisors, that is 1 for each micro processor system, in the same way as they monitor the DC-to-DC converters. In case of a fault, the supervisors signal this condition to the user logic by setting a status bit in order to warn the system of possible inconsistent inputs.

⚠ DANGER

LOSS OF THE ABILITY TO PERFORM SAFETY FUNCTIONS

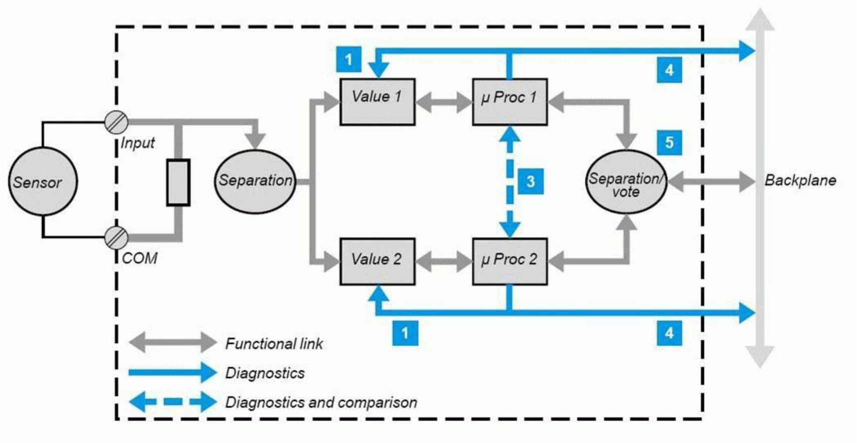
Use the correct process power supply, which is a PELV type with a maximum output of 60 V.

Failure to follow these instructions will result in death or serious injury.

Safety Analog Input Module

Architecture

The following figure shows the architecture of the Quantum Safety Analog Input module:



Legend:
µ Proc Microprocessor

The interface on the process side consists of 8 independent, isolated, current input channels.

Each input is acquired by 2 identical circuits:

1. The measuring devices are regularly monitored for their ability to measure, without a detected error, 5 analog values between 4 and 20 mA.
   The linearity of the measuring stages is verified at the same time.

3. to **5.** These mechanisms are described in the digital input module *(see page 47)* **140 SDI 953 00S** section.

Wiring Information

In order to ensure appropriate shielding characteristics of the wiring, you must use grounding equipment for the analog input shielded wires.

Schneider Electric recommends using the following devices from the Advantys STB catalog (MKTED206061EN) or similar equipment:

- Grounding Kit, part number STB XSP 3000
- Terminals for Grounding Kit, part number STB XSP 3010 or STB XSP 3020

Unused inputs are signaled as unhealthy because of the Safety analog input module's open circuit detection. The health bit of unused inputs should be masked in the health word of the module in your application logic.

Usage in Fire and Gas Applications

In fire and gas applications, the Safety analog input modules must be monitored for ground faults (leakage of current). The wires should be connected potential-free. With a shunt resistor (for instance 250 Ω) between the ground rail of the grounding kit and the earth ground, a voltage can be measured in case of a leakage of the current on 1 of the analog inputs. This voltage must be supervised to detect a leakage.

Description of the Diagnostics

The field side consists of 8 isolated independent input channels. Each input is acquired by 2 identical circuits. Each micro processor drives its ADC through isolators to acquire the input value. Further, it drives each DAC and may set it to high impedance (non-interfering) or low impedance, forcing the input of the ADC during diagnostics.

Diagnostic Timing

The analog input module performs:

- A short term (every 15 ms) self-test during normal, cyclical acquisition to detect a discrepancy that could result from an internal fault.
- An intermediate (every 18.75 s) self-test during diagnostic acquisition to verify the health of each channel.
- A long term (< 8 hours) self-test of the systems
- A power-on self-test which includes a full diagnostic of the process side (takes1.8 s) and of the system side (takes 25 s)

Description of the Power Supply Supervision

There is no power supply supervisor. This function is checked during ADC diagnostics as both ADC and DAC provide values dependant on their power supply voltage value.

Description of the Health Conditions

A module is healthy if the following tests are passed successfully:

- At each module cycle, the 2 systems cross check their behavior (state, data and timing consistency)
- Comparison of both Dual Port Memory data at each PLC access
- Clock frequency check (long term test)
- Memory check (long term test)
- Controller check (long term test)

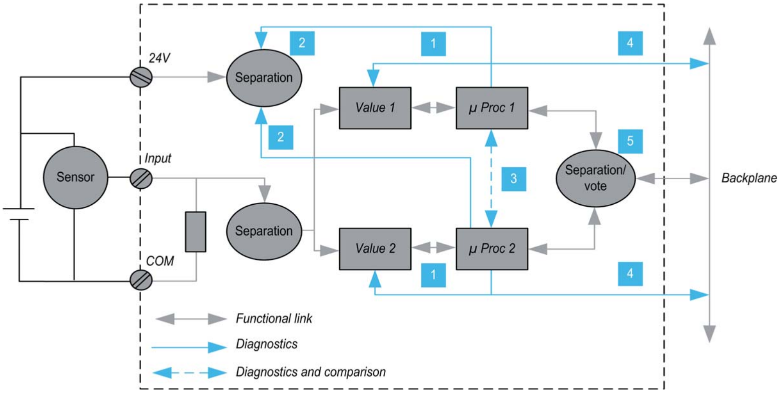
A channel is declared healthy if the module is healthy and the following tests are passed successfully:

- Every 15ms, the 2 systems compare the sampled measure
- Once every 250 module cycles, the measure is done by one system, while the other system checks its input (it forces 1 analog value out of 5 possible in the full input range and measures it). The full diagnostic is finished when all 5 values have been tested.

Safety Digital Input Module

Architecture

The following figure shows the architecture of the Quantum Safety Digital Input module:



Legend:
μ Proc Microprocessor

Each input channel uses a unique interface circuit and 2 independent inputs.

The diagram above shows that, except for the input terminal block screw and the backbone connection, the module is internally fully redundant. The input is connected to 2 different measuring devices, each controlled by a microprocessor.

The +24 V sensor supply voltage is also supplied to each of the 2 measurement channels, where its validly is tested. Each microprocessor stores data, then checks that the 2 measuring systems have worked correctly before sending the data to the PLC processor.

To do this cross-checking, each microprocessor:

1. Forces 0 and 1 on the measuring system, reads these values and then verifies that they are consistent with the 0 an 1 levels.
2. Verifies the presence of the +24 V that is needed to validate the measurement
3. Verifies that the other microprocessor has complied with the diagnostic and measurement protocols.
   The microprocessors then exchange data and compare their measurement results. Each then defines its response for the CPU by preparing a secure response frame with:
   - Time-based date
   - Identification of the module and its address
   - 32-bit CRC for reliable transmission

     Note: The maximum length of a data frame is 160 bits. The CRC/frame length ratio is such that the risk of non-detection of transmission errors on the assembly is almost zero.

4. The supply voltage from the backplane is monitored. If there is an over- or under-voltage from the backplane, the module goes into a safe fallback position.
5. For each input, both measurement channels must send the same data to the CPU. This is verified by the Vote function, which eliminates any risk of data degradation between the microprocessor stage and the backplane.

Wiring Information

NOTE: The Safety digital inputs are de-energized to trip. The Safe input state is the de-energized state, that means if the input state goes to de-energized, the Safety Function is executed. Therefore, the wiring must be done accordingly.

Connect the unused input channels of the used input modules to 24 VDC. This is required to avoid creating open circuit faults due to the Safety digital input module's open circuit detection of these unused inputs.

Description of the Diagnostics

Each input channel uses a common input circuit and 2 independent acquisition chains. Each micro processor drives a digital input serializer (DIS), which samples the input information. It also drives a digital input deserializer (DID) on each input circuit, which in turn drives the diagnostic block to set the diagnostic cases. The acquisitions are synchronous so that they can be compared.

Diagnostic Timing

The digital input module performs:

- A power on self-test which includes a full diagnostic of the process side (takes 5.1 s) and of the system side (takes 25 s)
- A short-term self-test (every 15 ms) during normal, cyclical acquisition to detect a discrepancy that could result from an internal fault.
- A intermediate self-test (every 60 ms) during diagnostic acquisition to verify the health of each channel.
- A long term self-test (< 8 hours) of the systems

Description of the Input Channel Error Detection

The digital input monitors the field side power supply. The external wiring is checked by sensing the leakage current. The minimum leakage current is 1mA. If there is no leakage current, this is detected as an open circuit. In case of dry contact, a pull-up resistor of 15 kΩ is needed to avoid broken wire detection. Each input circuit is equipped with switches, which are periodically forced to 1 or to open circuit state in order to check if the circuit is healthy. Each input circuit is checked independently and declared unhealthy if there is a detected fault by setting a diagnostic bit.

Description of the Health Conditions

A healthy module successfully passes the following tests:

- During each module cycle, the 2 systems cross check their behavior (state, data and timing consistency)
- Comparison of both Dual Port Memory data at each PLC access
- Clock frequency check (part of the long term test)
- Memory check (part of the long term test)
- Controller check (part of the long term test)
- During the continuous checking, the process power supply is checked by 2 independent circuits

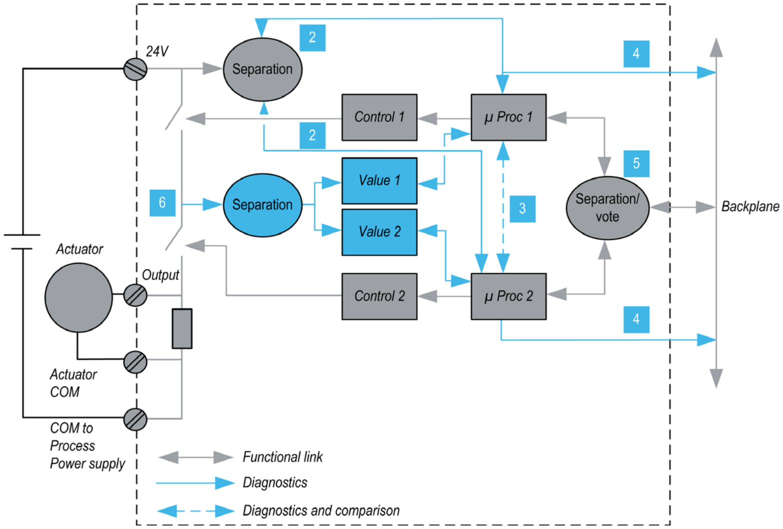
A channel is declared healthy if the module is healthy and the following tests are passed successfully:

- At each input sample (every 15ms), the 2 systems check the consistancy of the sample measurment
- Once every 4000 module cycles, the mesurement is done by only one System, while the other system checks that its input circuit is OK (it forces a 0 then a 1, then it checks the feedback value)

Safety Digital Output Module

Architecture

The following figure shows the architecture of the Quantum Safety Digital Output module:



Legend:
µProc Microprocessor

Diagnostic mechanisms 2 to 5 are identical to the discrete input module *(see page 47)* 140 SDI 953 00S mechanisms.

Like all safety modules and CPUs, the140 SDI 953 00S module is internally fully redundant. The output is controlled by 2 different control devices, which are controlled by a microprocessor.

The output stages are checked using a safety function.

Each output consists of 2 switches in series between the external +24 V power supply and the ground. The mid-point value (6) is read and sent to each microprocessor.

NOTE: The power supply is designed to detect any interruption of the voltage supplied to the output stage.

NOTE: These brief changes to “0“ have no effect on industrial control of motors or valves that are insensitive to these very short control disturbances.

⚠ WARNING

RISK OF UNINTENDED OPERATION

When the demand state of the output is energized, the monitoring of the health status of each channel output of the SDO module is mandatory to take an appropriate measure in accordance with the application.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Wiring Information

You should protect the field power supply of the Safety digital output modules by a fuse. This fuse protects the module not only against reversed field power supply, but also against field power supply overvoltage. There must not be any current limitation, and the field power supply must be able to deliver 50 A during 0.2 s in case of short circuit. The fuse must be chosen according to the driven load and must not exceed 16 * 0.5 * 1.25 = 10 A fast-blow (IEC 61131-2). Thus, you should use a fast 10 A fuse on the field power supply input of each digital output module.

⚠ WARNING

SHORT-CIRCUIT RISK

Use a 10 A, 250 V fast-blow fuse to protect the field power supply against reversed power and overvoltage.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

NOTE: The 10 A value for the fuse is an absolute maximum value. Adapt it to a lower value depending on the real consumption of the actuators and on the number of outputs used.
For example, if only 12 outputs are connected to actuators with an input current less than 0.2 A, the fuse value is 12*0.2 = 2.4 A < Fuse <= 10 A.

The ground of the field power supply must be wired to the terminal block of the output module separately from the ground of the actuators. For further details, see the Quantum with Unity Pro Discrete and Analog I/O Reference Manual.

NOTE: It is recommended connecting at least 2 ground lines (common 0 V) to the terminal block. For further details, see “Wiring Diagram” in Quantum with Unity Pro Discrete and Analog I/O Reference Manual

Diagnostic Timing

The digital output module performs:

- A power on self-test which includes a full diagnostic of the process side (takes 3.06 s) and of the system side (takes 25 s)
- A short-term self-test (every 15 ms) during normal, cyclical acquisition to detect a discrepancy that could result from an internal fault

- An intermediate self-test (every 24 s) during diagnostic acquisition to verify the health of each channel
- A long term self-test (< 8 hours) of the systems

Description of the Timeout State

Configure a timeout state for the Safety-Related output modules in the following cases:

- Detection of a malfunction of the CPU
- Occurrence of a communication problem
- Configure the following 3 timeout states:
  - Hold last value
  - User defined 0, which is the Safe state
  - User defined 1

For a detailed procedure for configuring the timeout state and the module timeout of the digital output modules, refer to the Unity Pro XLS Software Operating Mode Manual Safety PLC Specifics.

⚠ WARNING
POSSIBLE LOSS OF THE ABILITY TO ENTER THE SAFE STATE
Configure a timeout state of 0 to allow the Safety digitial output modules to go into the Safe state. If the configured state = 1, the Safety modules may not go into the Safe state. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

⚠ WARNING
RISK OF UNEXPECTED BEHAVIOR
If the module detects an internal error, the output goes into the Safe (de-energized) state whatever the value defined for the time-out state. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

Description of the Module Time-out

Configure the module time-out value, which must be compliant with the PLC cycle time, the Hot Standby configuration (if HSBY is used) and the process Safety time. For more information, refer to the Process Safety Time *(see page 75)*.

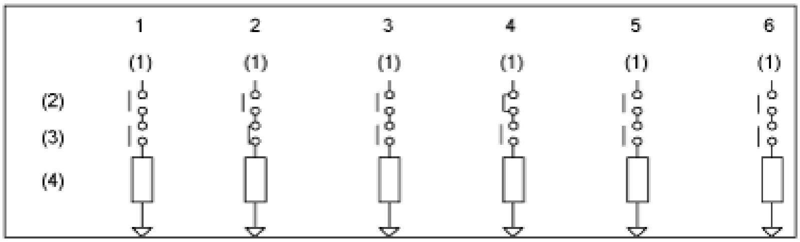
In case of permanent bad exchanges with the CPU, the digital output module reboots after a fixed time-out of 65 seconds. This causes all outputs to go to 0, independent of the configured time-out state.

Description of the Health Conditions

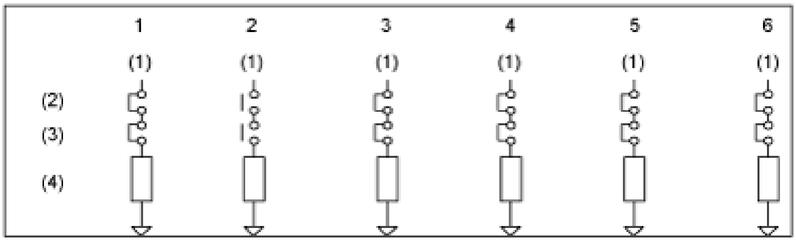
A healthy module successfully passes the following tests:

- During each module cycle, the 2 systems cross check their behavior (state, data and timing consistency)
- Comparison of both Dual Port Memories data at each PLC access
- Clock frequency check (part of the long term test)
- Memory check (part of the long term test)
- Controller check (part of the long term test)
- During the continuous checking, the process power supply is checked by 2 independent circuits
- During each module cycle, both systems check the consistency of CPU messages.
  A channel is declared healthy if the module is healthy and the following tests are passed successfully:
  - During each module cycle, both systems check the consistency state of the switches
  - Once every 1600 module cycles, both systems cooperate to perform a test of the "switch health". The sequence depends on the actual value of the output. The duration of each state is less than 1ms.

The circuits tested when output = 0:



The circuits tested when output = 1:



Legend:

1. 24 V process
2. High side switch
3. Low side switch
4. Load

Section 2.3
Power Supply

Power Supply for the Quantum Safety PLC

Introduction

For use in the Quantum Safety PLC, the 140 CPS 124 20 and 140 CPS 224 00 Quantum power supply modules are certified.

These CPS are certified, even if neither PFH nor PFD values are provided. They cannot deliver a higher voltage than the one supported by all the safety modules. For this reason, these CPS modules do not contribute to the global PFH or PFD values calculation for the safety function.

⚠ WARNING

LOSS OF THE ABILITY TO PERFORM SAFETY FUNCTIONS

Do not use power supply modules other than the Quantum 140 CPS 124 20 or 140 CPS 224 00.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Description of the Module Features

These modules have an over voltage protection and detection. Further, they support redundancy. If a fault is detected, the other one will take over and supply the rack with the necessary power.

NOTE: Schneider Electric recommends always using 2 Quantum power supply modules per rack in the Quantum Safety PLC.

NOTE: 1 power supply module (140 CPS 124 20 or 140 CPS 224 00) must be able to deliver the complete power of the drop.

For details of the configuration of modules, see *Configuration Examples for the Quantum Safety PLC, page 62*.

Section 2.4
Non-Interfering Modules

Non-Interfering Modules for the Quantum Safety PLC

Introduction

The Quantum Safety PLC consists of Safety modules that perform Safety functions. This PLC also supports **non-interfering modules**.

There are 2 groups of **non-interfering modules**:

- Modules needed to complete a Safety PLC:
  - 140 XBP 006 00 (Backplane 6 slots)
  - 140 XBP 010 00 (Backplane 10 slots)
  - 140 XBP 016 00 (Backplane 16 slots)
  - 140 CRP 932 00 (Remote I/O Head Adapter)
  - 140 CRA 932 00 (Remote I/O Drop Adapter)
  - 140 CPS 124 20 (Power Supply)
  - 140 CPS 224 00 (Power Supply)
  - 140 NRP 954 00 (Multi-mode Fiber Optic Repeater Module)
  - 140 NRP 954 01C (Single Mode Fiber Optic Repeater Module)

- Modules for additional non-safety functions:
  - 140 NOE 771 11 (Ethernet Module)
  - 140 DDI 353 00 (Digital Input)
  - 140 DDO 353 00 (Digital Output)
  - 140 ACI 040 00 (Analog Input)
  - 140 ACO 020 00 (Analog Output)
  - 140 ERT 854 20 (Time Stamp Module)

Additional parts such as cables and terminal strips are also available for a Quantum Safety PLC.

Description of the RIO Adapters

The RIO head adapter 140 CRP 932 00 and the RIO Drop Adapter 140 CRA 932 00 are allowed to be used for the communication between the Safety CPU and the Safety remote I/Os. For detailed information on this topic, see *Description of the CPU-I/O Communication, page 39*. All standard components of Schneider Electric for wiring remote I/Os (cables, connectors, and so on) are allowed to be used in the Safety-Related System.

Description of the Ethernet Module

The Ethernet module 140 NOE 771 11 can be used for the communication of the Safety PLC with other PLCs, HMIs or I/Os on the Ethernet network. It does not alter Safety-Related data and therefore is not part of the Safety loop. For detailed information on this topic, see *PLC-PLC Communication Description, page 108*. The Ethernet module can only be configured in the local rack.

Description of the Backplanes

The backplanes 140 XBP 016 00, 140 XBP 010 00, and 140 XBP 006 00 are the equipment on which you can mount all Safety and non-interfering modules.

NOTE: Backplane expanders are not allowed in the Quantum Safety PLC.

Description of the I/O and Time Stamp Modules

You are allowed to configure non-interfering I/O and time stamp modules in your Safety PLC. However, they must not be part of the Safety loop.

> ⚠ CAUTION
>
> INCORRECT USE OF SAFETY-RELATED DATA
>
> Make sure that neither inputs nor outputs of non-interfering I/O and time stamp modules are used for calculating Safety-Related outputs. These modules are only allowed to process non-Safety signals. The logic used to process the non-interfering I/Os must follow the same rules as for Safety logic. The non-interfering I/Os must be mapped to the Safety memory range.
>
> **Failure to follow these instructions can result in injury or equipment damage.**

The Unity PRO XLS cannot check this rule, so the user is responsible for the separation of safety logic and non-safety logic. It is recommended to use separate sections to facilitate the verification.

Section 2.5
Restrictions on I/O Modules

Description of the Restrictions on I/O Modules

Introduction

With regard to the communication between the Quantum Safety CPU and I/O modules, you must observe the following restrictions on I/O modules:

- Communicating to I/O via Ethernet or Modbus Plus is not allowed on a Quantum Safety PLC. Unity Pro XLS cannot check for compliance with this rule because Ethernet and Modbus Plus communication to other PLCs (not I/Os) are allowed, see also PLC-PLC Communication Description *(see page 108)*.

⚠ DANGER

UNCERTIFIED DATA TRANSFER – SIL3 VIOLATION

Do not configure Ethernet or Modbus Plus I/Os in your Safety PLC. It is your responsibility to guarantee that no communication occurs to I/O via Ethernet or Modbus Plus. Any violation of this rule makes your application non-IEC 61508 compliant.

Failure to follow these instructions will result in death or serious injury.

- Distributed I/Os, which communicate via Modbus Plus, are not allowed in the Quantum Safety PLC. Unity Pro XLS checks that no distributed I/Os are configured. If you do not obey this rule, the Unity Pro analyzer does not generate code.
- I/Os communicating via other fieldbuses are not allowed in the Quantum Safety PLC. Unity Pro XLS checks that no fieldbus I/Os are configured. If you do not obey this rule, the Unity Pro analyzer creates a relevant error message and does not generate code.

Section 2.6
System Behavior in Case of Detected Diagnostic Errors

Introduction

The Safety CPU modules and the Safety I/O modules have internal diagnostics to check if the modules are working correctly. This chapter describes the behavior of the modules in case an error is detected. Also, your possibilities to intervene are explained.

What Is in This Section?

This section contains the following topics:

Improper Behavior of the Safety CPU Modules

General

The CPU diagnostics verifies the correctness of the hardware and the running program, see *Standalone Safety CPU, page 33*. If an error is detected during 1 of the tests, the CPU enters an error state and all Safety-related outputs go to the Safe state.

Handling Detected Errors

If a an error is detected, perform the following steps:

Step	Action
1	Power off the complete PLC.
2	Switch the power on again. **Result:** A self-test is performed.
3	Read the content of the system words %SW125, %SW126, and %SW127 for information on the detected error state, see *Description of the System Words %SW60 to %SW127, page 168*.
4	Provide the contents of these system words and Unity Pro project system words to Schneider Electric support.

Some of the detected errors are temporary and disappear after a restart of the PLC. Others require replacement of the CPU.

NOTE: If an **"Automatic Start in Run"** option for the CPU is configured (its use is not recommended in a Safety PLC) and if the diagnostic error is persistent, the CPU again enters the error state and stops.
To read the values of the system words, prevent a restart by either:

- removing the PCMCIA memory card (the application is stored on the card)
- by inserting an empty PCMCIA memory card (the application is stored in memory)

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Avoid using the **Automatic start in Run** option. If you use this feature, it is your responsibility to program and configure the system in such a way that it behaves correctly after restart.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Content of the System Words

%SW125 contains the cause of the detected error and have the following meaning:

Code (hex)	Meaning
0x5AF1	sequence check detected error (unpredictable execution in CPU)
0x5AF2	detected error in memory (incorrect address)
0x5AF3	detected comparison error (result of the execution of the Intel processor differs from that of the application processor)
0x5AF4	real-time clock detected error
0x5AF5	detected error initializing double code execution
0x5AF6	detected watchdog activation error
0x5AF7	detected error during memory check (it takes more than 8 hours)
0x5AF8	detected error in memory check (in RAM)

%SW126 and %SW127 contain information that is for Schneider Electric internal use to analyze the problem in more detail.

Improper Behavior of the Safety I/O Modules

General

The Safety I/O modules detect an internal error in either:

- a channel
- the complete module

Detected Channel Error

If an error is detected in a channel, this channel is set to the Safe state while the other channels continue to operate. The information about the detected error is available in the status registers of the module (see "Quantum Safety I/O Modules" in the *Quantum with Unity Pro Discrete and Analog I/O Reference Manual*). Depending on the type of detected error, the complete module may have to be exchanged.

Detected Module Error

If a module error is detected, the I/O module enters the Safe state. It then resets, restart sand performs the power up self-tests:

If the power up self-tests ...	Then the module ...
are successful	starts and operates normally.
are unsuccessful	resets and goes through the same procedure. **NOTE:** If several self-tests are unsuccessful, the module must be exchanged.

After a detected error in a Safety I/O module, it restarts automatically. If the power-up self tests are successful, the module continues normal operation, i.e., it again sets the outputs to 1. If an inoperable module has been exchanged (hot-swapped), it also automatically starts operation after the self-test. The application must be programmed and configured in such a way that it behaves correctly after restart of the Safety I/O modules.

⚠ WARNING

UNEXPECTED APPLICATION BEHAVIOR - AUTOMATIC RESTART

Program and configure the system in such a way that it behaves correctly after the Safety I/O modules restart.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Section 2.7
Configuration Examples

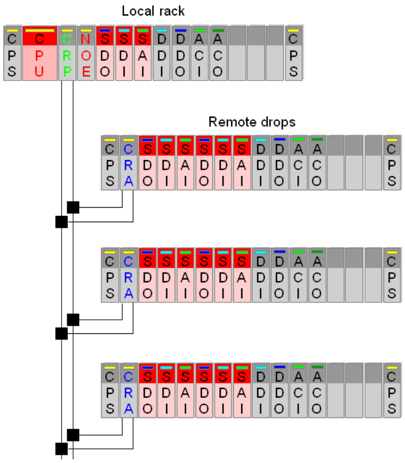
Configuration Examples for the Quantum Safety PLC

Introduction

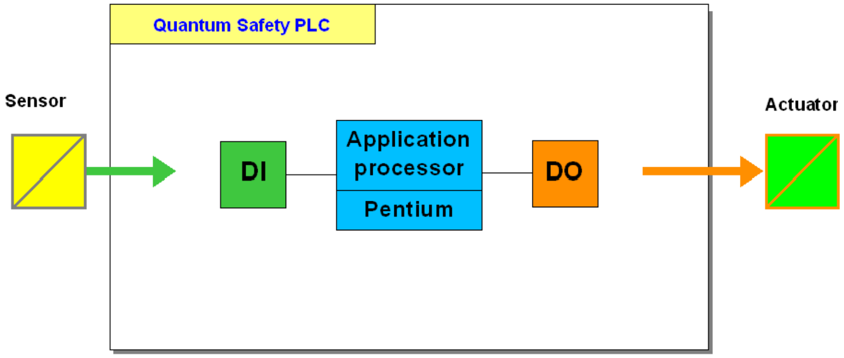
The Quantum Safety PLC can consist of a local rack and additional remote I/O drops. All Safety modules on the local and remote racks are in the safety loop. The Safety PLC and the Safety I/O modules can be configured as either non-redundant or redundant.

Standalone Configuration (1oo2 HotStandby system)

The following is an example of a standalone Quantum Safety PLC, consisting of a local rack and 3 remote I/O drops:



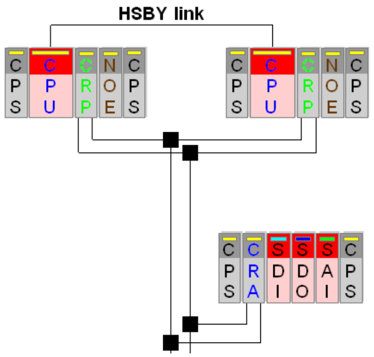
The following figure provides the appropriate functional overview:



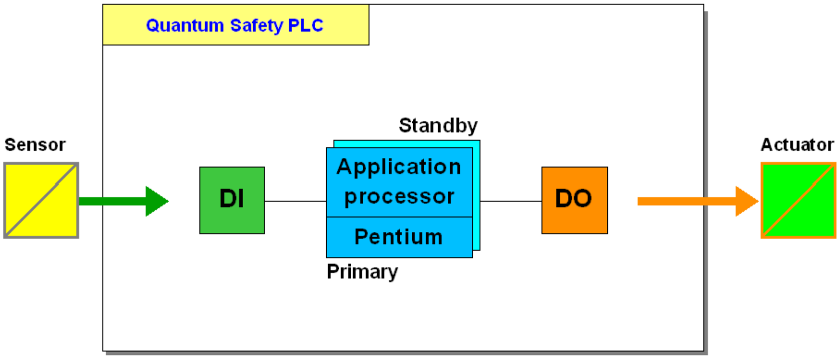
NOTE: Only the Schneider Electric Safety products are certified for use in your Quantum Safety PLC and therefore for processing Safety-Related data. Non-interfering modules such as the Ethernet NOE communication module are only certified for processing non-Safety-Related data. However, they are allowed to be part of the Quantum Safety PLC because they cannot interfere with the Safety-Related System by their own means. Still, they are not allowed to execute Safety Functions. Further, you can connect other necessary devices such as human-machine interfaces (HMI). These devices are not part of the Safety loop because they are not allowed to write Safety data directly, see also *PLC-HMI Communication Description*, page 124.

Redundant CPU Configurations for High Availability (1oo2 HotStandby system)

The following figure is an example of a Hot Standby Quantum Safety PLC consisting of redundant CPUs:

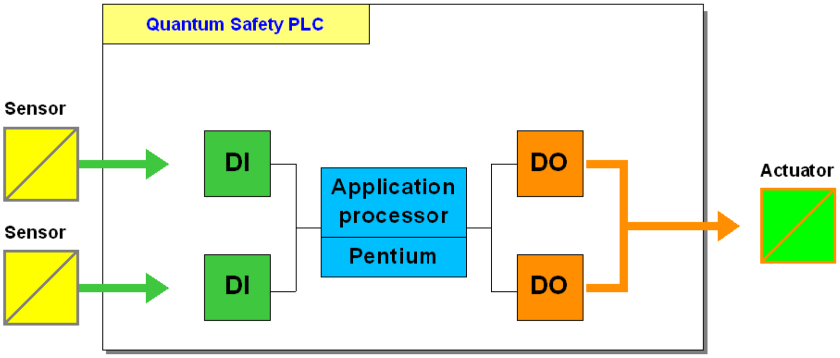


The following figure provides the appropriate functional overview:



Redundant I/O Configurations for High Availability

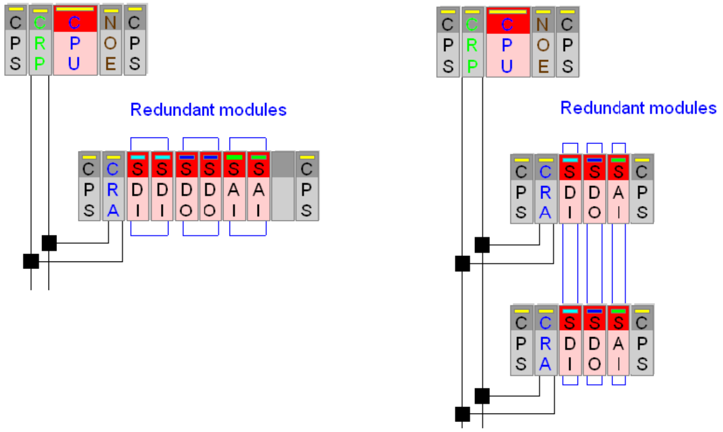
The following figure provides the functional overview of a redundant I/O configuration, consisting of 1 CPU and redundant I/Os:



It is possible to place your redundant Safety I/O modules

- either in the same RIO drop (not recommended)
- or in different RIO drops (recommended when redundant Safety I/O modules are used).

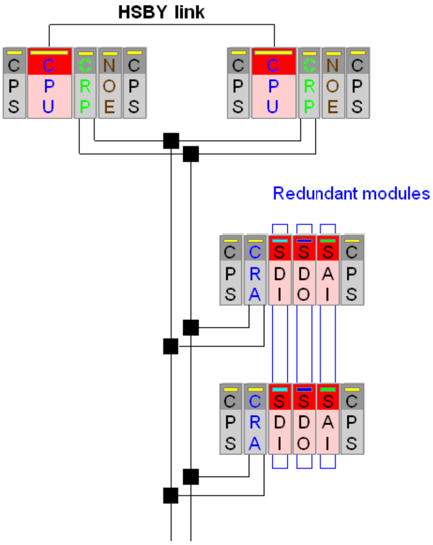
The following figure shows redundant I/Os placed in the same RIO drop (left) and in different RIO drops (right):



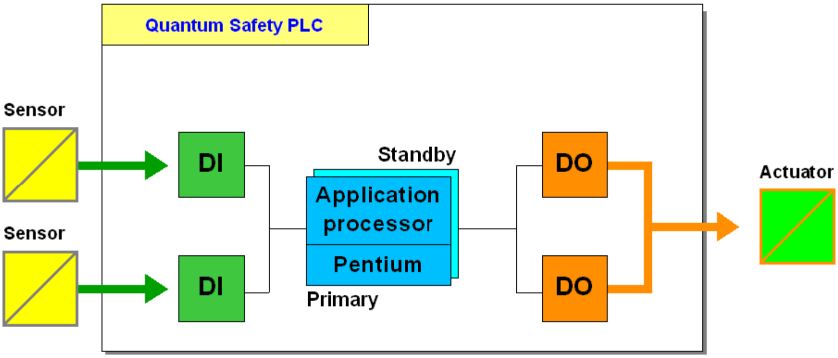
NOTE: Schneider Electric recommends always placing redundant Safety I/O modules in different RIO drops.

Redundant CPU and I/O Configuration

The following figure shows an example of a Quantum Safety PLC consisting of redundant CPUs and redundant I/Os:



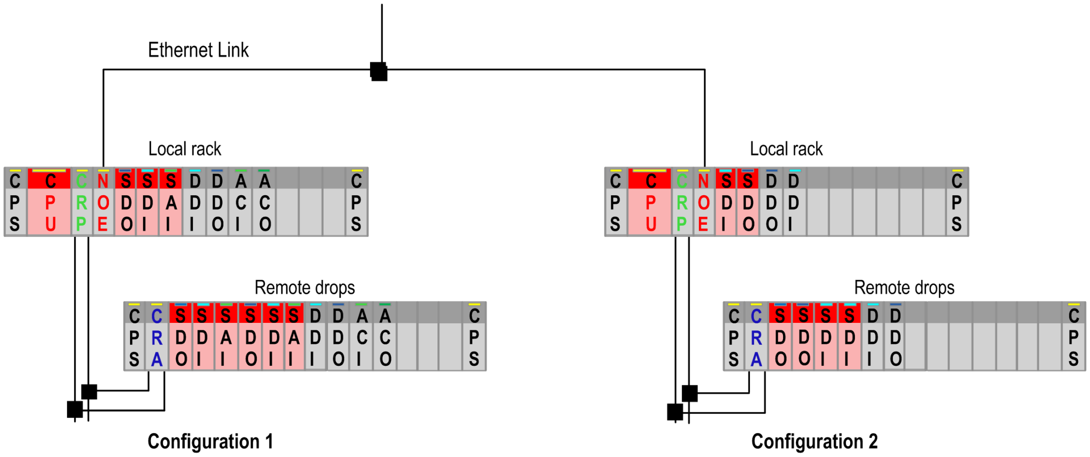
The following figure provides the appropriate functional overview:



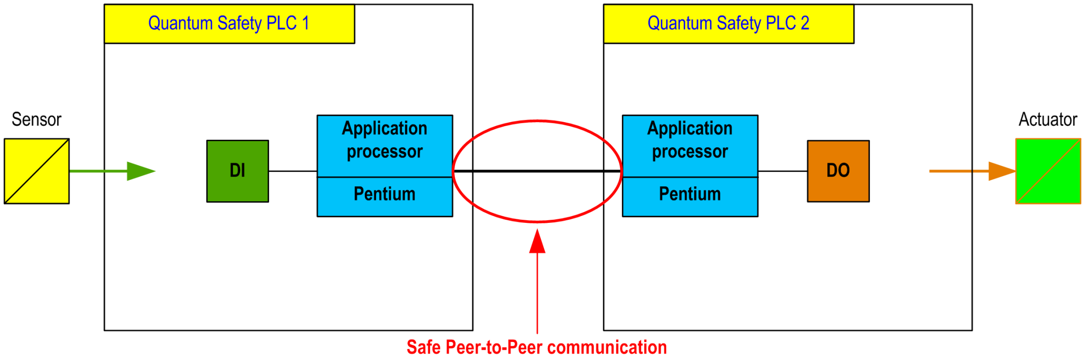
NOTE: Schneider Electric recommends always placing redundant Safety I/O modules in different RIO drops.

Peer-to-Peer Standalone Configuration

The following figure shows an example of peer-to-peer standalone Quantum Safety PLC, consisting of two standalone configurations which communicate throughout a black channel *(see page 120)* on an Ethernet link:



The following figure provides the appropriate functional overview:



Chapter 3
Programming

Introduction

This chapter deals with the topics important for programming your SIL3 project. The requirements for programming a Safety-Related System are described and the SIL3 features are explained.

What Is in This Chapter?

This chapter contains the following sections:

Section 3.1
General Information on Programming

Introduction

This section provides general information on programming a SIL3 application with regard to programming and monitoring requirements.

What Is in This Section?

This section contains the following topics:

Available Language Sections

Introduction

For programming your SIL3 project, you are only allowed to use the following 2 programming languages:

- function block diagram (FBD)
- ladder diagram (LD)

Both are languages defined by the IEC 61131-3 for the programming of PLCs.

Description of the Restrictions on Language

If you create a SIL3 project, the following restrictions apply:

- At creation time, Unity Pro XLS restricts your choice of programming language.
- At import time, Unity Pro XLS ignores any section other than FBD or LD, but does not stop the import. The use of sections other than FBD or LD generates errors.
- At analyze time, Unity Pro XLS checks each section for its language. If any test fails, it creates an error and does not generate your program.

You can find a detailed description of the restrictions on program structure, language elements, and data configuration in *Exceptions and Requirements for Programming, page 72*.

Exceptions and Requirements for Programming

Introduction

To program a SIL3 project, you must use the programming languages FBD and LD only while at the same time observing the rules listed below concerning the program structure, language elements, and data configuration.

Requirements for the Program Structure

You are only allowed to program your SIL3 project in master task (MAST task) sections.

You are not allowed:

- to program **FAST**, **TIMER**, **INTERRUPT**, and **AUX** tasks. In case of an import, Unity Pro XLS ignores the objects not allowed and informs you of their existence. If you continue the import, it is done without the objects that are not allowed, which may lead to errors or it may stop if the import is not possible
- to use subroutines (SR sections)
- to schedule segments
- to call remote I/Os in parallel

⚠ **WARNING**

POSSIBLE LOSS OF THE ABILITY TO PERFORM SAFETY FUNCTIONS

Do not use conditional section execution with Unity Pro XLS.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Requirements for Language Elements

You are only allowed to use functions and function blocks (FFBs) that are certified for use in Safety logic and described in the Unity Pro Safety Block Library.

You are allowed to create your own derived function blocks (DFBs) and store them in the safety library.

You are not allowed to use ST expressions.

In LD, you are not allowed to use:

- halt coils
- call coils
- returns
- operate blocks
- compare blocks

NOTE: Though jumps to labels are allowed in FBD and LD, Schneider Electric recommends not using them for a better structuring of your Safety logic.

Requirements for Configuring Data

You are only allowed to use:

- the elementary data types (EDTs) `BOOL`, `EBOOL`, `BYTE`, `WORD`, `DWORD`, `INT`, `UINT`, `DINT`, `UDINT`, `FLOAT` and `TIME`
- simple arrays (the index can only be a literal), for details see the chapter "Programming" in the *Unity Pro XLS Operating Mode Manual Safety PLC Specifics*
- direct addressing, for instance, writing %MW4000 by a coil in LD
- located variables. All instances of variables are not only checked with regard to being located but also as to being located in a valid memory area, see also *Memory Area Description, page 104*

You are not allowed to create derived data types (DDTs).

NOTE: You are not allowed to use variables from the unrestricted memory areas in your user logic unless you may connect it to the input of `S_SMOVE_BIT` or `S_SMOVE_WORD` function blocks, see also *Memory Area Description, page 104*.

Checks for Programming

At creation time of a SIL3 project, Unity Pro XLS offers only the features allowed for Safety logic. Any attempt to create objects not allowed leads to an error.

However, objects not allowed can be inserted through source file import. Therefore, Unity Pro XLS checks all objects at analyze time. At any rule not obeyed or any object not allowed, Unity Pro XLS creates an error and does not generate your project.

In the project settings, Unity Pro XLS provides the following different options concerning the warnings of the language analyzer:

- Variables not used
- Multiple writing of variables
- Parameters not assigned
- Multiple use of FB instances
- Overlapping of addresses

⚠ WARNING
POSSIBLE LOSS OF THE ABILITY TO PERFORM SAFETY FUNCTIONS Switch on all warning options in the project settings and check the warnings to make sure that they are not critical and that the behavior is intended. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

Requirements for Monitoring

Unity Pro XLS is the only programming software allowed to load or to modify your SIL3 project. Other programming packages or HMIs may monitor both the state and functions of the Safety-Related System but must not alter them. Any other device is allowed to read data from the Safety PLC but writing to a Safety PLC is restricted, see also *Memory Area Description, page 104*.

Process Safety Time

Description of the Process Safety Time

The process Safety time (PST) is a critical measure of each process. It is defined as the period between the occurrence of a failure in equipment under control (EUC) and the occurrence of a hazardous event if the Safety Function is not performed.

NOTE: The process Safety time is given by the process. It must be ensured that the Safety-Related System is able to perform the Safety Functions within the process Safety time.

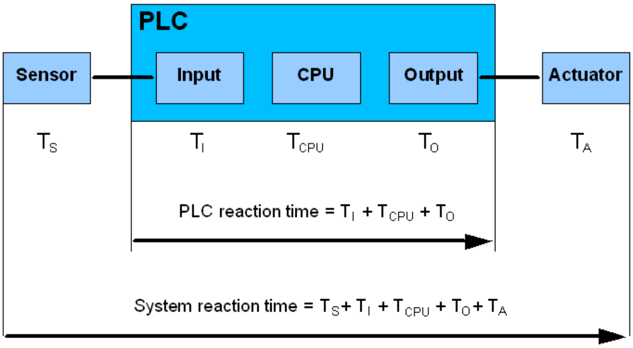
Description of the System Reaction Time

The system reaction time is the sum of the PLC reaction time and the time for the used sensor device (T_S) and the time for the used actuator device (T_A). T_S and T_A are device specific.

The following equation is valid:

System reaction time = PLC reaction time + T_S + T_A

This equation is illustrated below:


PLC
Sensor
Input
CPU
Output
Actuator
T_S
T_I
T_{CPU}
T_O
T_A
PLC reaction time = $T_I + T_{CPU} + T_O$
System reaction time = $T_S + T_I + T_{CPU} + T_O + T_A$

The system reaction time must be less than the process Safety time.

Description of the PLC Reaction Time

The PLC reaction time is the sum of the related time for the used input module (T_I) and the used output module (T_O) and the CPU reaction time (T_{CPU}).

The following equation is valid:

PLC reaction time = $T_{CPU} + T_I + T_O$

Description of the CPU Reaction Time

The CPU reaction time is directly impacted by the CPU cycle time which is needed to execute the Safety logic. A signal may appear just at the beginning of the execution cycle when the signals have already been processed. Therefore, 2 cycles may be necessary to react to the signal.

This leads to the following equation:

CPU reaction time = 2 x CPU cycle time

In addition, it is possible to define a maximum number of accepted CRC faults (N_{CRC}) for the communication with the I/Os. This has been introduced to reduce spurious effects (for instance by an EMC disturbance). This number can be defined to take a value between 1 and 3. This must be taken into account because the number of cycles for the output module to react is increased.

Therefore, the equation above is extended as follows:

CPU reaction time = $(2 + N_{CRC})$ x CPU cycle time

NOTE: If you are using a peer-to-peer safe communication to perform the safety function, the CPU reaction time estimation is different *(see page 120)*.

Description of the Time for Input Modules

The maximum times (worst case) for the Safety digital input module and for the Safety analog input module T_I are 45 ms (3 times the module's cycle time).

Description of the Time for Output Modules

The maximum time T_O for the Safety digital output module is equal to the cycle time of the module:

T_O = 15 ms

For the Safety digital output module, a timeout T_{OUT} must be configured. The module timeout must be greater than the CPU cycle time, see below.

You can find a detailed procedure for configuring the module timeout of digital output modules in the chapter "Configuring I/O Modules for Safety Projects" in the *Unity Pro XLS Operating Mode Manual Safety PLC Specifics*.

Description of the Maximum CPU Cycle Time

Knowing the required PST and the maximum reaction time of the sensors and actuators, you are able to calculate the maximum PLC reaction time tolerable in your process.

To ensure that the system reaction time is smaller than the process Safety time, the maximum CPU cycle time must fulfill the following condition:

Max. CPU cycle time < $(PST - T_I - T_O - T_S - T_A) / (2 + N_{CRC})$

In addition, you must consider the following relation between the maximum timeout T_{OUT} for the output modules and the maximum CPU cycle time:

T_{OUT} > max. CPU cycle time x $(1 + N_{CRC})$

NOTE: If you are using a peer-to-peer safe communication to perform the safety function, the maximum CPU cycle time estimation is different *(see page 120)*.

Example Calculation

The following values are given:

- required PST = 1.1 s
- T_I = 45 ms
- T_O = 15 ms
- T_S = 100 ms
- T_A = 500 ms
- N_{CRC} = 1

The maximum CPU cycle time is calculated as follows:

Max. CPU cycle time < (1100 ms - 45 ms - 15 ms - 100 ms - 500 ms) / 3

Max. CPU cycle time < 146.7 ms

The requirement that the module timeout of the digital output module must be greater than the CPU cycle time is fulfilled:

T_{OUT} > 300 ms

In case of a fault of the CPU, the outputs are set to Safe state after the timeout has expired. Therefore, the system needs the following time to shut down the outputs:

$T_{OUT} + T_O$

In the example, this time amounts to

300 ms + 15 ms = 315 ms

CPU Cycle Time in a Hot Standby System

In a normally running Hot Standby system, the formula for the CPU cycle time is the same:

Max. CPU cycle time < $(PST - T_I - T_O - T_S - T_A) / (2 + N_{CRC})$

In addition, you must consider the following relation between the maximum timeout T_{OUT} for the output modules and the maximum CPU cycle time:

T_{OUT} > 4 x max. CPU cycle time (worst case)

Configuring the Maximum CPU Cycle Time

The Quantum Safety PLC can perform cyclic or periodic execution. There is no difference between the behavior of a standard Quantum PLC and a Quantum Safety PLC regarding cyclic and periodic execution. In both cases, you must configure the maximum acceptable CPU cycle time in Unity Pro XLS.

The maximum allowed CPU cycle time (watchdog) is configured in the properties of the MAST task. For details, see the chapter "Programming" in the *Unity Pro Operating Modes Manual* and the chapter "Presentation of the Master Task" in the *Unity Pro Program Languages and Structure Reference Manual*.

NOTE: The minimum CPU cycle time is 20 ms.

NOTE: Only configure a maximum number of %M and %MW that is really needed. All configured memory ranges %M and %MW are compared as part of the double execution, which takes roughly 5.5 ms per 10,000 words. Therefore, you increase the cycle time unnecessarily if you configure more memory than needed.

You must check your CPU cycle time when commissioning your project. At this time, Unity Pro XLS provides the real time values from the PLC.

You can find this information

- in the **Task** tab available using the menu entry **Tools → PLC Screen**.
- in %SW30, containing the current time of the MAST task execution.
- in %SW31, containing the maximum time of the MAST task execution.
- in %SW32, containing the minimum time of the MAST task execution.

For details, see *Description of the System Words %SW30 to %SW59, page 164* or the chapter "Description of the System Words %SW30 to %SW47" in the *Unity Pro Program Languages and Structure Reference Manual*. If your maximum acceptable CPU cycle time is exceeded, you must adjust your configuration or your user logic or both to reach the required value.

⚠ WARNING

RISK OF EXCEEDING THE PROCESS SAFETY TIME

Set the maximum CPU cycle time taking into account your process Safety time.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Section 3.2
Software Description

Introduction

This section describes the special characteristics of Unity Pro XLS developed to program SIL3 applications.

What Is in This Section?

This section contains the following topics:

Unity Pro XLS

Introduction

To meet the requirements of the IEC 61508, only certified software is allowed for programming SIL3 applications. For this purpose, Schneider Electric has developed the Safety version of the programming tool Unity Pro XLS (XL-Safety). It is able to perform both fault diagnostics and project protection to an extent necessary for programming a SIL3 project.

NOTE: When you create a new project with Unity Pro XLS, the choice of the Quantum PLC type determines if a SIL3 or non-Safety project is created.

SIL3 and Non-Safety Applications

Unity Pro XLS can be used to program both SIL3 and non-Safety applications. Thus, no other programming software is necessary. Only 1 version can be installed on your computer.

Your SIL3 project is stored in binary project files (*STU*) and in archive project files (*STA*). You cannot open these files with non-Safety versions of Unity Pro. Further, you can only download your executable binary files (*APX*) into a Safety CPU. For details, see the chapter "Services in Offline Mode" in the *Unity Pro Operating Modes Manual*.

Non-Safety projects created by non-Safety Unity Pro versions must be exported using the appropriate Unity Pro version and imported into Unity Pro XLS.

Description of the Project Protection

Unity Pro XLS offers protection against unauthorized access concerning your SIL3 project and the Quantum Safety PLC as well as Unity Pro XLS itself.

Your SIL3 project and the Quantum Safety PLC are protected by the following password mechanisms:

- The SIL3 project is protected by a password at the application level, the application password. When you create a SIL3 project, an empty password, which you can change, is set.
- The Quantum Safety PLC is also protected by the application password. In case there is no application in the PLC, it accepts any password.
- Connecting to a Safety PLC requires to enter the application password if the currently opened project in Unity Pro XLS is different or no project is opened.

Unity Pro XLS itself is protected by the following mechanisms:

- You can define access rights or a list of functions a user is allowed to perform using the Security Editor provided together with Unity Pro XLS (and having the same functionality as in Unity Pro XL).
- After a configured time of inactivity, Unity Pro XLS is locked automatically. Before being able to continue to work with it, you must enter the application password. While Unity Pro XLS is locked, the connection to the PLC is maintained and it stays in the current mode.

Description of the Security Editor

To protect Unity Pro XLS against unauthorized access, you can use the Security Editor

- to apply a policy and to create profiles and users for it.
- to manage access rights to it.

For example, you can restrict the access for

- creating or modifying the application password,
- entering Maintenance Mode, or
- adapting the auto-lock timeout.

For details of using the Security Editor, see also the chapter "Access Security Managment" in the *Unity Pro Operating Modes Manual* and the chapter "Security Management for Unity Pro XLS" in the *Unity Pro XLS Operating Mode Manual Safety PLC Specifics*.

NOTE: Use the features provided by the Security Editor to protect Unity Pro XLS against unauthorized access. However, using the Security Editor does not remove the necessity to protect your SIL3 project by using an application password.

Description of the Auto-Lock Feature

Unity Pro XLS offers the option to protect itself against unauthorized access after a configured time of inactivity. After this time is exceeded, Unity Pro XLS prompts you to enter the application password.

You can find a detailed procedure for activating the auto-lock in the chapter "Protection of a Safety Project with Unity Pro XLS" in the *Unity Pro XLS Operating Mode Manual Safety PLC Specifics*.

Default Values

When you create a new SIL3 project, the following values are set by default:

- The application password is empty.
- The auto-lock is activated, allowing 10 minutes of inactivity before Unity Pro XLS is locked.

Functions/Function Blocks for SIL3 Applications

Introduction

Schneider Electric offers you a number of elementary functions (EF) and function blocks (EFBs) that are certified for use in SIL3 applications. For details, see the Unity Pro Safety Block Library.

Remark

FFBs that are available for different data types are labeled with ***.

For example, the elementary function S_AND_*** is available for the data type

- BOOL as S_AND_BOOL.
- BYTE as S_AND_BYTE.
- WORD as S_AND_WORD.
- DWORD as S_AND_DWORD.

Description of the Safety FFBs for Mathematics

The following table lists the Safety FFBs belonging to the family of mathematic functions:

Name	Type	Used...
S_ADD_***	EF	to add the input values
S_SUB_***	EF	to substract the input 2 from the input 1 value
S_MUL_***	EF	to multiply the input value
S_DIV_***	EF	to divide the Divident input value by the Divisor input value
S_NEG_***	EF	to negate the input values
S_ABS_***	EF	to compute the absolute value of the input value
S_SIGN_***	EF	to detect negative signs
S_SMOVE_BIT	EFB	to assign the input value to the output (to use data from unrestricted memory area in the Safety logic)
S_SMOVE_WORD		

Description of the Safety FFBs for Comparison

The following table lists the Safety FFBs belonging to the family of comparison functions:

Name	Type	Used to check the values of successive inputs...
S_EQ_***	EF	for equality
S_GT_***	EF	for a decreasing sequence
S_GE_***	EF	for a decreasing sequence or equality
S_LT_***	EF	for an increasing sequence
S_LE_***	EF	for an increasing sequence or equality
S_NE_***	EF	for inequality

Description of the Safety FFBs for Logic

The following table lists the Safety FFBs belonging to the family of logic functions:

Name	Type	Used...
S_AND_***	EF	to perform a bit by bit AND link of the input bit sequence
S_OR_***	EF	to perform a bit OR link of the input bit sequence
S_XOR_***	EF	to perform a bit XOR link of the input bit sequence
S_NOT_***	EF	to negate the input sequence bit by bit
S_SHL_***	EF	to shift a bit pattern to the left
S_SHR_***	EF	to shift a bit pattern to the right
S_ROL_***	EF	to rotate a bit pattern circularly to the left
S_ROR_***	EF	to rotate a bit pattern circularly to the right
S_RS	EFB	as RS memory with a dominant reset input
S_SR	EFB	as SR memory with a dominant set input
S_F_TRIG	EFB	to detect falling edges
S_R_TRIG	EFB	to detect rising edges

Description of the Safety FFBs for Statistics

The following table lists the Safety FFBs belonging to the family of statistical functions:

Name	Type	Used...
S_MIN_***	EF	to assign the smallest input value to the output
S_MAX_***	EF	to assign the largest input value to the output
S_LIMIT_***	EF	to transfer the unchanged input value to the output if it lies within the minimum and the maximum limit
S_MUX_***	EF	to transfer the respective input value to the output depending on the K input value
S_SEL	EF	for a binary selection between 2 input values

Description of the Safety FFBs for Timers and Counters

The following table lists the Safety FFBs belonging to the family of timer and counter functions:

Name	Type	Used...
S_CTU_***	EFB	for counting upwards
S_CTD_***	EFB	for counting downwards
S_CTUD_***	EFB	for counting upwards and downwards
S_TON	EFB	as on delay timer
S_TOF	EFB	as off delay timer

Name	Type	Used...
S_TP	EFB	for generating a pulse with defined duration

Description of the Safety FFBs for Type Conversion

The following table lists the Safety FFBs belonging to the family of type conversion functions:

Name	Type	Used to convert an input value of the data type...
S_BOOL_TO_***	EF	BOOL to a BYTE, WORD, DWORD, INT, DINT, UINT, or UDINT data type
S_BYTE_TO_***	EF	BYTE to a BOOL, WORD, DWORD, INT, DINT, UINT, or UDINT data type
S_WORD_TO_***	EF	WORD to a BOOL, BYTE, DWORD, INT, DINT, UINT, or UDINT data type
S_DWORD_TO_***	EF	DWORD to a BOOL, BYTE, WORD, INT, DINT, UINT, or UDINT data type
S_INT_TO_***	EF	INT to a BOOL, BYTE, WORD, DWORD, DINT, UINT, or UDINT data type
S_DINT_TO_***	EF	DINT to a BOOL, BYTE, WORD, DWORD, INT, UINT, or UDINT data type
S_UINT_TO_***	EF	UINT to a BOOL, BYTE, WORD, DWORD, INT, DINT, or UDINT data type
S_UDINT_TO_***	EF	UDINT to a BOOL, BYTE, WORD, DWORD, INT, DINT, or UINT data type

Description of the Safety FFBs for High Availability

The following table lists the Safety FFBs belonging to the family of functions for high availability:

Name	Type	Used...
S_DISIL2	EFB	to select the data from the 2 digital input modules in case of a redundant input module configuration
S_AISIL2	EFB	to select the data from the 2 analog input modules in case of a redundant input module configuration

Description of the Safety FFBs for Hot Standby

The following table lists the Safety FFBs belonging to the family of functions for Hot Standby:

Name	Type	Used...
S_HSBY_SWAP	EFB	to swap between primary and standby CPU in case of a Hot Standby solution

Details on how to use the Safety FFBs in your project are provided in the *Unity Pro Safety Block Library*.

Description of the Safety DFBs for Safe Peer-to-Peer Communication

The following table lists the Safety DFBs belonging to the family of functions for Safe Peer-to-Peer communication:

Name	Type	Used...
`S_WR_ETH`	EFB	to compute data to send on Safe Peer-to-Peer communication from the sender PLC
`S_RD_ETH`	EFB	to compute data received from Safe Peer-to-Peer communication in the receiver PLC

Details on how to use the Safety DFBs in your project are provided in the *Unity Pro Safety Block Library*.

Description of the Non-Interfering FFBs for Quantum I/O

⚠ **WARNING**

LOSS OF ABILITY TO PERFORM SAFETY FUNCTIONS

Do not use output of the non-interfering function blocks to perform Safety functions.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

The following table lists the non-interfering FFBs belonging to the family of functions for Quantum I/O:

Name	Type	Used...
`NI_QUANTUM`	EFB	to edit the configuration data of a Quantum main rack
`NI_DROP`	EFB	to edit the configuration data of a remote I/O station
`NI_ERT_854_20`	EFB	to give access to the time stamp and status functions of the 140 ERT 854 20 module

Details on how to use the non-interfering FFBs in your project are provided in the *Unity Pro Safety Block Library*.

Application Password

Password Protection Management

In the following situations, you are requested to enter the application password:

- opening an existing SIL2 or SIL3 project
- modifying the application password
- clearing the application password
- connecting to the Safety PLC
- exceeding the configured time of inactivity and launching the auto-lock mechanism

You can find detailed procedures for managing the application password in the chapter "Project Properties and Password for Unity Pro XLS" in the *Unity Pro XLS Operating Mode Manual Safety PLC Specifics*.

NOTE: Schneider Electric strongly recommends changing the default password immediately after having selected a Quantum Safety CPU in order to protect your project against unauthorized access from the beginning. Yet, if you forget to change the default password, the empty password is kept even if you save and close your project. When re-opening it, just click **OK**, that is leave the edit field empty, and change the password as soon as possible.

Losing the Application Password

You can find detailed procedures for what to do in case you have lost the application password in the chapter "Loss of Password" in the *Unity Pro XLS Operating Mode Manual Safety PLC Specifics*.

Section 3.3
Operating Procedures

Introduction

This section deals with the operating procedures of the Quantum Safety PLC with special regard to its 2 special operating modes.

What Is in This Section?

This section contains the following topics:

Operating Modes of the Safety PLC

Introduction

The default behavior of the Quantum Safety PLC is to perform Safety Functions to achieve and to maintain the Safe state of a process. Nevertheless, you must be able to debug and to maintain your project.

Use the Safety Mode to control your process and the Maintenance Mode for debugging and refining your project.

In Maintenance Mode, the I/O and CPU modules are still executing the diagnostics and establishing the Safe state if a fault is detected. Only the application program and the application data, which may be changed in Maintenance Mode, are not checked.

NOTE: To program a Safety PLC, Unity Pro XLS is required.

Safety and Maintenance Mode Features

The operating mode of the Quantum Safety PLC depends on events such as application exception, power on/off, and so on. The functions available in Unity Pro XLS depend on the operating mode.

Switching between the modes requires defined conditions and follows certain procedures. For details, see the chapter “Switching Between Safety and Maintenance Mode” in the *Unity Pro XLS Operating Mode Manual Safety PLC Specifics*.

You can interact with the Safety PLC using:

- Unity Pro XLS programming tool
- Quantum Safety CPU keypad
- Quantum Safety CPU key switch

Depending on the operating mode, the Safety PLC can be in different states.

After power up, it automatically enters run state of the Safety Mode if the following 2 conditions are fulfilled:

- There is a valid application.
- The **Automatic start in Run** option is activated.

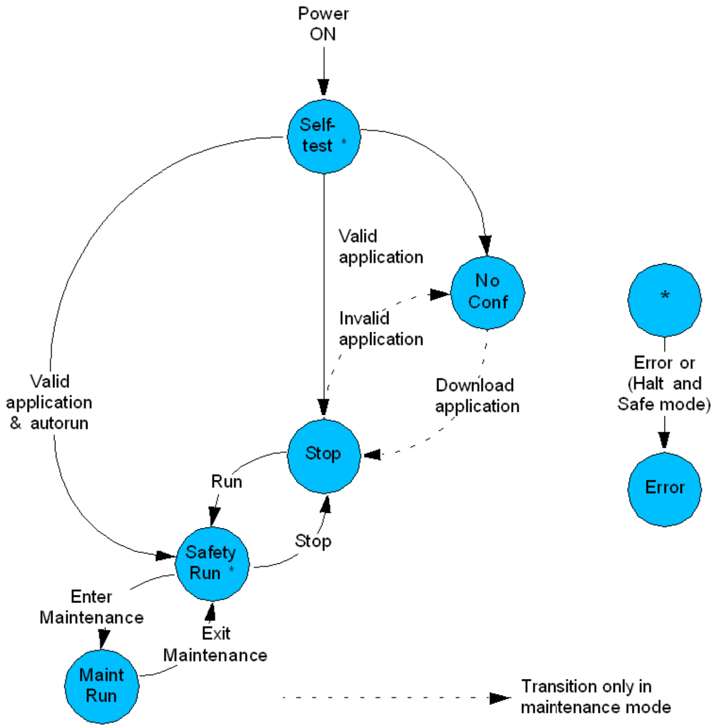
In case of an invalid application, it enters the not configured (no conf) state of the Maintenance Mode (only if the key state is unlocked), in which you are able to download your project.

If a fault is detected, the PLC enters

- Halt state when running in Maintenance Mode.
- Error state when running in Safety Mode.

PLC States

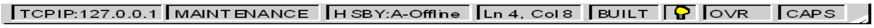
The following figure shows the state diagram of the Quantum Safety PLC:



Operating Mode Identification

The LCD display on the CPU indicates the current operating mode by showing the letters *M* for Maintenance Mode or *S* for Safety Mode.

The status bar field on the PLC screen indicates the current operating mode as shown in the following figure:



TCPIP:127.0.0.1 | MAINTENANCE | HSBY:A-Offline | Ln 4, Col 8 | BUILT | OVR | CAPS

Safety Mode

Safety Mode Description

The Safety Mode is the default mode of the Quantum Safety PLC. It is a restricted mode in which modifications and maintenance activities are prohibited.

Safety Mode Restrictions

When the PLC is running in Safety Mode, the following restrictions are implemented by Unity Pro XLS:

- Download changes are not allowed.
- Setting and forcing of Safety variables and Safety I/Os is not allowed.
- Debugging with breakpoints, watch points, and single step is not allowed.
- Animation tables and operator screens must not write Safety variables and Safety I/Os.
- The Safety memory is write protected; that means that human-machine interfaces (HMIs) and other PLCs cannot write to it. This is controlled by the Safety PLC, see also *Memory Area Description, page 104*.

NOTE: The logic animation, animation tables, and operator screens can influence the scan time.

NOTE: It is possible to download a new version of the Ethernet processor firmware into the Quantum Safety CPU with the OSLoader. However, it is only allowed to do that in Maintenance Mode.

⚠ WARNING

POSSIBLE LOSS OF THE ABILITY TO PERFORM SAFETY FUNCTIONS

Do not download a new version of the Ethernet processor firmware into the Quantum Safety CPU in Safety Mode. It is possible to do so but not allowed.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Safety Mode States

Once the Safety Mode is entered, the PLC can be in run state and in error state. In run state, all restrictions are active and the results of the double user code execution are compared. If any test is unsuccessful, the PLC goes to error state because it has no means to recover from the error.

Entering Safety Mode

There are 4 ways of entering Safety Mode:

- when the Safety PLC is powered up
- when the Safety Mode is entered from Maintenance Mode
- when the key is locked
- when Unity Pro XLS is disconnected either by the customer or because of a broken connection

When the Safety PLC is powered up, it automatically enters Safety Mode.

NOTE: After power up and if there is a valid application, the PLC only performs cold start. Thus, the project is reinitialized and the system performs:

- the initialization of data with the initial values defined in the project
- the initialization of elementary function blocks (EFBs) based on initial data
- the initialization of data declared in the EFBs
- the initialization of system bit and words
- the cancellation of any forcing, see also *Forcing, page 94*

Switching from Maintenance Mode to Safety Mode is only possible if the PLC is not debugging.

NOTE: Data forced before switching to Safety Mode stay forced after switching, see also *Forcing, page 94*.

Details concerning the transition from Maintenance Mode to Safety Mode can be found in the chapter "Switching Between Safety and Maintenance Mode" in the *Unity Pro XLS Operating Mode Manual Safety PLC Specifics*.

Automatic Start in Run Option

You have the possibility to let your project automatically enter Safety Mode’s run state after power up. To do this, activate the option **Automatic start in Run**, see also the chapter "Configuration of Quantum Processors" in the *Unity Pro Operating Modes Manual*. However, Schneider Electric recommends using the **Run** command instead of the **Automatic start in Run** option for a SIL3 project to enter run state.

⚠ WARNING
UNINTENDED EQUIPMENT OPERATION
Avoid using the **Automatic start in Run** option. If you use this feature, it is your responsibility to program and configure the system in such a way that it behaves correctly after restart.
Failure to follow these instructions can result in death, serious injury, or equipment damage.

Maintenance Mode

Maintenance Mode Description

The Maintenance Mode of the Quantum Safety PLC is a temporary mode for modifying your project and debugging and maintaining your program.

Maintenance Mode Features

This mode is available in both RUN and STOP.

The PLC allows direct transition from RUN SAFE to RUN MAINTENANCE (and RUN MAINTENANCE to RUN SAFE.)

The Maintenance Mode (protected by a password) allow users to perform:

- online modifications (programs enhancements, temporary modifications, etc.)
- forcing values for sensor or actuator maintenance
- system installation and commissioning

When the PLC is running in Maintenance Mode, the following features are implemented by Unity Pro XLS:

- Download changes are allowed.
- Setting and forcing of Safety variables and Safety I/Os is allowed. However, only variables of the type `EBOOL` can be forced.
- Switching to Safety Mode while forcing is allowed. The forced variables stay forced, see also *Forcing, page 94*.
- Debugging with breakpoints, watch points, and single step is allowed. However, the PLC must be in run state.
- Animation tables and operator screens can write Safety variables and Safety I/Os.
- The Safety memory is write protected; that means that HMIs or other PLCs cannot write to it. This is controlled by the Safety PLC, see also *Memory Area Description, page 104*.

Entering Maintenance Mode

You can only enter Maintenance Mode from Safety Mode because after power up the PLC automatically enters Safety Mode. To exit Safety Mode and enter Maintenance Mode, the key switch must be unlocked. You can find procedures for switching between the modes in the chapter "Switching Between Safety and Maintenance Mode" in the *Unity Pro XLS Operating Mode Manual Safety PLC Specifics*.

Maintenance Mode States

In Maintenance Mode, the PLC can be in run state or in halt state. When it is in run state, you can modify your project. Further, you can switch to Debug Mode if you want to debug and maintain your program. In run state, the double code execution is performed but the result of the comparison is ignored.

⚠ DANGER

RISK OF LOSING THE SAFETY FUNCTION DURING COMMISSIONING AND MAINTENANCE

All modifications of the running system must follow the requirements of the IEC 61508.

Failure to follow these instructions will result in death or serious injury.

Forcing

Introduction

Forcing is only possible in Maintenance Mode. However, it is possible to switch from Maintenance Mode to Safety Mode while data are forced and the forcing stays active.

NOTE: Check the latest version of the TÜV document *Maintenance Override* for the procedures which must be applied when using forcing in a Safety-Related System. You can find it on the TÜV Rheinland Group website *http://www.tuvasi.com/*.

⚠ WARNING

LOSS OF ABILITY TO PERFORM SAFETY FUNCTIONS

Make sure that the forcing is turned on only temporarily and that the user logic is supervising the status of forcing (%SW108, see *Description of the System Words %SW60 to %SW127, page 168*).

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Handling Forced Data

Because forced data stays forced, Unity Pro XLS warns you before executing your switch command from Maintenance Mode to Safety Mode and prompts you to confirm it.

NOTE: In case of a disconnection between PLC and Unity Pro XLS, the latter also warns you if there are forced data, independently of the mode the PLC is in. This is due to the fact that the PLC automatically enters Safety Mode when being disconnected from Unity Pro XLS by the user or a communication interruption.

⚠ WARNING

RISK OF PROCESSING FORCED DATA

Check the state of your data before switching from Maintenance Mode to Safety Mode. Forced data stays forced and the PLC continues processing them. Make sure that your PLC processes the correct, unforced data necessary for performing the Safety Functions.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

You can check the status of forcing by reading the system word %SW108. It contains the number of forced I/O module bits. The system word is incremented on every forcing and decremented on every unforcing.

Section 3.4
Special Features and Procedures

Introduction

This section explains the special features and procedures using Unity Pro XLS as a programming tool for SIL3 projects.

What Is in This Section?

This section contains the following topics:

Checking the Programming Environment

Introduction

Unity Pro XLS provides the possibility to perform a self-test in order to verify that the components currently in use are the correct versions originally installed and are not corrupted, for instance by hard disk corruption. The self-test is done by evaluating the CRC.

Description of the Self-Test

When performing the self-test, Unity Pro XLS checks the version and CRC of

- DLLs of Unity Pro XLS,
- the Safety FFB-library database, and
- the hardware catalog database.

⚠ **WARNING**

RISK OF CORRUPTED PROGRAM

Use the self-test of Unity Pro XLS on a regular basis to check the integrity of your program. At least, perform the self-test

- after installing any software on or removing it from your computer.
- before loading the final operating program into the Safety PLC.
- before modifying a program in the running Safety PLC.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

You can find details of how to start the self-test in the chapter "Unity Pro XLS Self-Test" in the *Unity Pro XLS Operating Mode Manual Safety PLC Specifics*.

Starting the Quantum Safety PLC

Preconditions

Preconditions to start the Quantum Safety PLC are that you have

- configured your Safety-Related System correctly,
- programmed your SIL3 project correctly,
- tested the integrity of both your SIL3 project and Unity Pro XLS,
- connected Unity Pro XLS to your Safety PLC, and
- downloaded your SIL3 project into the Safety PLC.

Starting the Quantum Safety PLC

Once the Quantum Safety PLC contains a valid project, it only performs cold start. Therefore, you can only start your SIL3 project by performing a cold start except when you have just downloaded your project into the PLC.

Hence, you can start your project out of the following 2 initial states:

- The PLC is powered up and you have downloaded your SIL3 project since power up.
- The PLC is powered off.

Further, Unity Pro XLS offers the **Automatic start in Run** option. If it is activated, your PLC automatically enters run state in Safety Mode after power up. However, Schneider Electric recommends not using this option.

⚠ **WARNING**

UNINTENDED EQUIPMENT OPERATION

Avoid using the **Automatic start in Run** option. If you use this feature, it is your responsibility to program and configure the system in such a way that it behaves correctly after restart.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

You can find detailed procedures for starting a SIL3 project in the chapter "Starting and Stopping a Safety Project" in the *Unity Pro XLS Operating Mode Manual Safety PLC Specifics*.

Version Stamp

Version Stamp Description

In Unity Pro XLS, each generated binary file of a SIL3 project has a version stamp, providing date and time of build. Thus, you can check both if and when your project has been subject to modifications.

You can find a detailed procedure for checking the project version in the chapter "Project Properties for Unity Pro XLS" in the *Unity Pro XLS Operating Mode Manual Safety PLC Specifics*.

Upload

Uploading a SIL3 Project

Uploading a project from the PLC to Unity Pro XLS is also available for SIL3 projects. If you want to use this feature, it must be selected in the project settings. In a SIL3 project, the application password must be known to be able to connect to the PLC. In addition, the PLC must be switched to Maintenance Mode to perform the upload. For further details, see the chapter "Project Settings" in the *Unity Pro Operating Modes Manual*.

Project Backups

Introduction

Unity ProXLS checks the integrity of your SIL3 project by calculating a CRC when you close it and checking the CRC when you open it again. The CRC indicates by changing its value if your project has been damaged or corrupted. In this case, the comparison indicates the values are not the same and Unity Pro XLS does not open your project. As a result, you cannot connect Unity Pro XLS to the Safety PLC and, therefore, have no possibility of modifying or repairing your corrupt project.

Project Backup Description

Besides uploading the project from the PLC (see *Upload, page 99*), the only way to get access to your project is to have a copy of its original, that is a backup of your project. From this backup, you can copy back your project data, that is restore them.

NOTE: Create backups of your SIL3 project on a regular basis. Once your project is damaged or corrupted, you cannot open it to modify or repair it yourself.

Advice for Creating Backups

Creating backups requires careful planning including consideration of

- Backup software
  Automated backup cannot be affected by human error to the extent that manual backup can.
- Backup procedure
  Making more than 1 copy and storing them offsite increases the possibility of a successful data recovery
- Backup type.
  In general, a backup can be full, incremental, or differential depending on which data it backs up.
- Backup interval
  Regularly scheduled backups improve the reliability of data recovery.
- Backup media type
  Whereas hard disk based storage is very practical, remote backups imply offsite storage.

Advice for Recovering Data

A backup is only as useful as its associated recovery strategy. Therefore, it is not only important to save the backup data but also to have access to the software required to read them.

NOTE: Choose the backup policy that is the most appropriate for your Safety-Related System. Make the adequate amounts and types of backups. Test frequently the process for restoring the original project from the backup copies.

Detected Faults

Introduction

If a fault is detected by any of the internal diagnostic measures and system tests, the behavior of the Quantum Safety PLC varies according to the mode that it is in.

Fault Behavior in Safety Mode

Running in Safety Mode, your PLC enters error state in case of a single detected fault because it has no means to recover from it. The error state is a hardware locked state. Your project is stopped and you cannot intervene or communicate with it.

Leaving the Error State

The only way to leave the error state is to start your PLC again, whereupon the PLC performs self-tests and initializes your project.

If your project ...	Then your PLC...
is valid	enters stop state, which it is forced to do because of the detected fault.
is invalid	enters no conf state.

NOTE: The PLC may be in an error state if the persistent detection of an error occurs. In this case, it may be necessary to replace the PLC.

Depending on the state which your PLC is in, perform the following steps:

If your project is ...	Then ...
in stop state and the autorun option activated	● either power on your PLC again ● or perform a **Run** command.
in stop state and the autorun option not activated	perform a **Run** command.
in no conf state	download a backup of your project.

Fault Behavior in Maintenance Mode

Running in Maintenance Mode, your PLC enters

- halt state in case of a diagnostic error.
- error state in case of a hardware watchdog occurrence.

If the PLC is in halt state, you still have the possibility to communicate with it and therefore to debug your project. With the **Init** command or the download of a project, the PLC goes to stop state and can now be restarted. If the PLC is in error state, the behavior is the same as described above in *Fault Behavior in Safety Mode*.

Chapter 4
Communication

Introduction

This chapter deals with the communication of the Quantum Safety PLC with Unity Pro XLS as well as with other devices.

What Is in This Chapter?

This chapter contains the following sections:

Section 4.1
Memory Area

Memory Area Description

Introduction

In Safety Mode, the Safety CPU rejects all write requests to the following memory areas:

- %M or %Q (0x register)
- %MW or %QW (4x register)
- EFB data

However, because it may be necessary for you to be able to write data to the Safety PLC, the memory is divided into a Safety and an unrestricted part, allowing you to write in %M as well as in %MW.

Safety Memory Description

The Safety memory area is write protected for any other device.

NOTE: The write access is controlled inside the CPU because some communications, for example with the HMI or with other PLCs (Safety or non-Safety), are not configured in the Safety PLC with Unity Pro XLS and therefore cannot be checked in the Unity Pro XLS configuration.

Write Protection Description

To prevent other devices from writing to the Safety memory area, there is a blocking mechanism. The PLC does not execute any write command and returns an error code.

Unrestricted Memory Area Description

The unrestricted memory area (UMA) is a specially dedicated memory area for bits and words which is not write protected. It has the following characteristics:

- It is located at the beginning of the complete memory range.
- Its size can be configured in Unity Pro XLS.
- Its values cannot be used directly but by using specific function blocks.

Configuring the Unrestricted Memory Area

You can configure the size of your unrestricted memory area in Unity Pro XLS in the CPU configuration with the following limits:

- In %MW, the limit is
  - the last word in the unrestricted area or
  - 0 if this area is not used.

- In %M, the limit is
  - a multiple of 16 and the last %M in the unrestricted area or
  - 0 if it is not used.

NOTE: Configure the unrestricted memory area first and confirm that the configured area is large enough. If this part of the memory must be modified later, all addresses must be changed.

⚠ CAUTION

RISK OF CORRUPT PROJECT

Check that the size of the unrestricted memory area is correctly stored in the Quantum Safety CPU after the download of the PLC application. To do so, you must read the system words %SW110 and %SW111 (for instance using the animation table) and compare them with the configured values in your application.

Failure to follow these instructions can result in injury or equipment damage.

Using Data from the Unrestricted Memory Area

To perform Safety Functions, you are only allowed to process data stored in the Safety memory area. If it is necessary to get access to the Safety Functions, you are allowed to use data from the unrestricted memory data. However, for Safety reasons, you cannot process them directly. Instead you must transfer data from the unrestricted memory area to the Safety memory area in order for Safety Functions to use these data.

You can find a detailed procedure for transferring data from the unrestricted to the Safety memory area in the chapter "Using Data from the Unrestricted Memory Area" in the *Unity Pro XLS Operating Mode Manual Safety PLC Specifics*.

Description of the Safety Move Function Blocks

Because you are not able to work with the values located in the unrestricted memory area directly, there are the following 2 function blocks enabling you to transfer data from the unrestricted memory area to the Safety memory:

- `S_SMOVE_BIT` to get access to bits
- `S_SMOVE_WORD` to get access to words

The variables from the unrestricted memory area are connected to the input of the function block, and its output is connected to a Safety variable. Direct addresses cannot be used because they are interpreted as `INT`. The WORD to be moved must be configured in the unrestricted memory area. If the actual value is not within the range, the output is set to 0 and the error is indicated. Additional inputs are used to control how the function blocks transfer the data to the outputs in case some data can only be used together in the same cycle.

NOTE: It is good practice to use an appropriate naming convention for variables from the unrestricted memory area and to comment them accordingly. This eases the audit of your SIL3 project.

The user can use data in the safety application by implementing a verification protocol (for example, send a word and its complement and then check the consistency in the application, copy the word in a new location and then reread the value, etc.).

Write Protection Description

Unity Pro XLS checks at edition time and at build time that only variables from the unrestricted memory area are used as input to the Safety MOVE function blocks. In addition, Unity Pro XLS provides a cross-reference feature to search for the variable usage, enabling you to check the rule easily.

⚠ WARNING

RISK OF PROCESSING INCORRECT DATA

Make sure that the data you move to the Safety memory area are correct data. Data transferred to the Safety memory area using the Safety MOVE function blocks are not automatically correct.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

To help ensure that data are transferred accurately, you can write the data to 2 different variables and then compare them.

Section 4.2
PC-PLC Communication

PC-PLC Communication Description

Introduction

Once you have programmed your SIL3 project, you must connect Unity Pro XLS to the Quantum Safety PLC if you want to download, run, and maintain it. To enable the communication between both, you can connect Unity Pro XLS to the following:

- Modbus TCP (either with CPU or NOE module)
- Modbus Plus
- Modbus RS232 / RS485
- USB

The communication between Unity Pro XLS and the Quantum Safety PLC is not part of the Safety loop but nevertheless subject to checks. For instance, a CRC is used during the download of a project in order to verify that the data are transferred correctly and that there is no communication error. However, you must additionally check the version and functionality of your project as well as the Unity Pro XLS environment.

For the Ethernet cabling, the standard Ethernet devices can be used.

Section 4.3
PLC-PLC Communication

PLC-PLC Communication Description

Introduction

Concerning a Safety PLC, only writing to other PLCs is allowed. Reading from other PLCs is only allowed in the unrestricted memory area, see also *Memory Area Description, page 104*.

NOTE: The write access is controlled inside the CPU because some communications, for example with the HMI or with other PLCs, are not configured in the Safety PLC with Unity Pro XLS and therefore cannot be checked in the configuration.

The Quantum Safety PLC is able to communicate with other PLCs using the following:

- Modbus TCP (either with CPU or NOE module)
- Modbus Plus
- Modbus RS232 / RS485

These kinds of communication are categorized as non-interfering.

NOTE: Communication from the Quantum Safety PLC as a Modbus Master via Modbus is not allowed because the function blocks are not certified. However, as a Modbus slave, the Safety PLC may be connected to other PLCs and communicate data when requested, or even accept data in the unrestricted memory area.

Description of the Ethernet Communication

The Ethernet network can be connected to

- either the Ethernet port of the CPU
- or the Ethernet module 140 NOE 771 11.

NOTE: In case of a Hot Standby Safety CPU, the Ethernet port is used for the data exchange between the primary and the standby CPU and therefore not available for the communication with other PLCs or HMIs.

The Ethernet module 140 NOE 771 11 is certified as non-interfering product for use in the Quantum Safety PLC. The communication can be either peer-to-peer or as global data.

For the Ethernet cabling, the standard Ethernet devices can be used.

Configuring the Ethernet Peer-to-Peer Communication

The peer-to-peer communication is configured in Unity Pro XLS in the Ethernet network configuration, independently for reading and writing. Unity Pro XLS checks that reading uses only the unrestricted memory area. It creates an error and does not generate code if this rule is not obeyed.

Configuring the Ethernet Global Data Communication

The global data communication is configured in Unity Pro XLS in the Ethernet network configuration to publish data for writing and to subscribe to data for reading. Because reading is only allowed from the unrestricted memory area, Unity Pro XLS checks this rule and creates an error if it is not obeyed.

Description of the Modbus Plus Communication

The Modbus Plus module 140 NOM 2XX 00 is not allowed for communication. You can only use the Modbus Plus port of the CPU. On the Modbus Plus network, a peer-to-peer communication or a global data exchange is possible.

Configuring the Modbus Plus Peer-to-Peer Communication

The peer-to-peer communication is configured in Unity Pro XLS in the Modbus Plus network configuration, independently for reading and writing. Unity Pro XLS checks that reading uses only the unrestricted memory area. It creates an error and does not generate code if this rule is not obeyed.

Configuring the Modbus Plus Global Data Communication

The global data communication is configured in Unity Pro XLS in the Modbus Plus network configuration, independently for reading and writing. Unity Pro XLS checks that reading uses only the unrestricted memory area. It creates an error and does not generate code if this rule is not obeyed.

⚠ WARNING

UNDETECTABLE LOSS OF DATA

Do not write from an external device to the Safety memory area in the Quantum Safety PLC using Ethernet. The data are ignored because of the Safety PLC’s write protection. The data are lost without you being notified.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Section 4.4
Safe Ethernet PLC-PLC Communication

What Is in This Section?

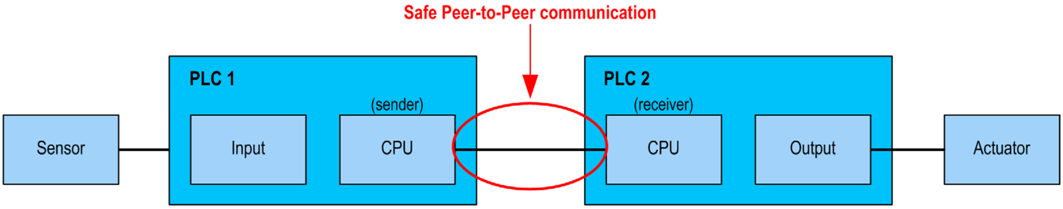
This section contains the following topics:

Peer-to-peer Communication

Introduction

By implementing a specific configuration, you are able to use Ethernet based peer-to-peer communication to perform the safety function with a SIL3 level.

The following figure provides the safety peer-to-peer communication functional overview:



This specific safe peer-to-peer Ethernet communication is based on a black channel. The protocol checks detected errors such as detected transmission errors, omissions, insertions, wrong order, delays, incorrect addresses, masquerade bits and manages retransmissions.

This safe communication is possible and allowed only between Quantum Safety PLCs.

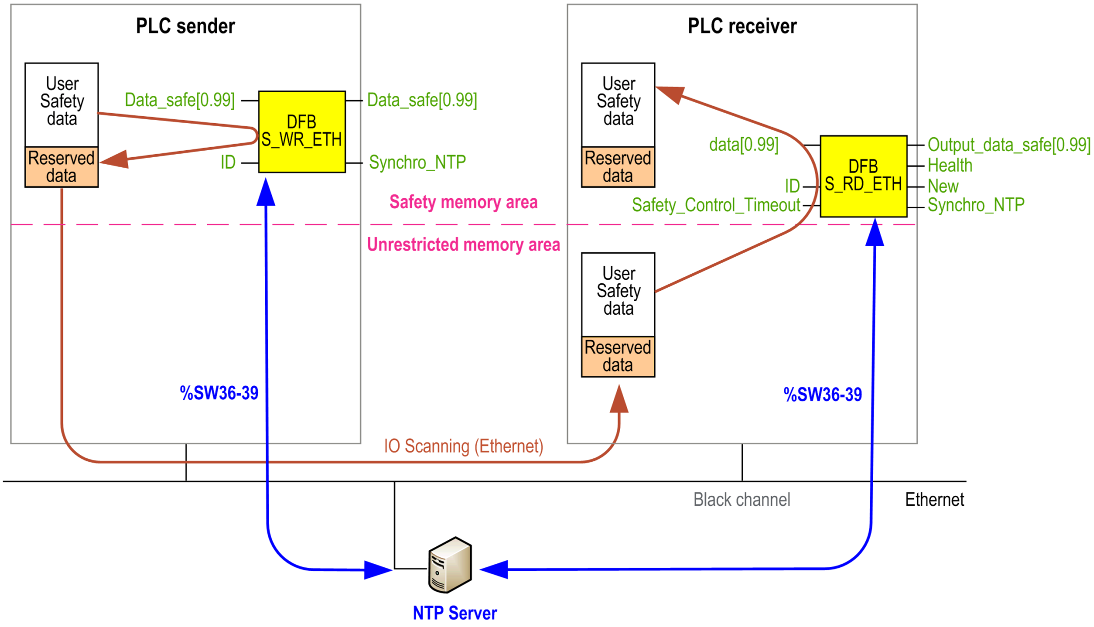
Solution Architecture

Description of the Configuration

The solution architecture is based on:

- NTP service for time base synchronization
- execution of 2 DFBs (`S_WR_ETH` in the sender PLC and `S_RD_ETH` in the receiver PLC)
- IO scanning service on Ethernet for data transportation (Modbus TCP)

The following figure shows the overview of the configuration required to establish the safe peer-to-peer communication:



On the Ethernet network, you are allowed to mix safety related data and non safety related data without impact on the integrity level of the safety related data.

There is no restriction on the Ethernet network when using the safe peer-to-peer communication. You must conform to the *Modicon Quantum with Unity Ethernet Network Modules User Manual*.

To connect a 140 NOE 771 11 module to the Ethernet network, Schneider Electric recommends to use the following switches:

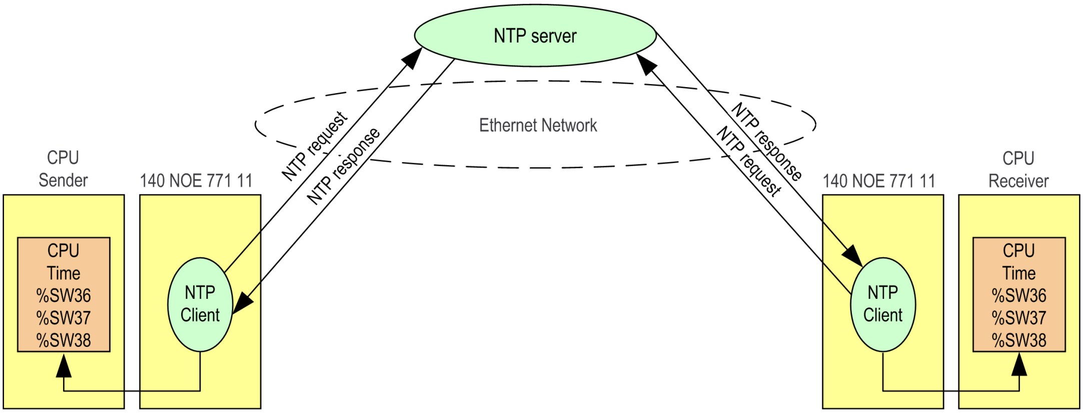
- 499NES17100
- 499NOS17100

Configuration of NTP Service

Description

The safe Ethernet PLC-PLC communication needs the synchronization of both PLCs (sender and receiver) time base. You have to configure the NTP service on each receiver and sender PLC by using the 140 NOE 771 11 non interfering module in Unity Pro *(see Modicon Quantum with Unity, Ethernet Network Modules, User Manual)*.

The following figure describes the sender and receiver PLCs time base synchronization principle:



The CPU operating system updates at each cycle some system words (`%SW36` to `%SW38`) that contain a time used by the safe communication as a time base. This time is internally filtered to avoid important time shifts and forbid fugitive bad values to be received from the NTP server.

The `%SW39` system word allows to diagnose the health of the time taken into account by the DFBs used in the user program in order to implement the safe peer-to-peer communication.

In Unity Pro, you must configure the NTP service parameters as follows:

- set the **Polling period** value to 20 seconds
- configure the same time zone for both sender and receiver PLCs in the **Time Zone** box
- uncheck the **Automatically adjust clock for daylight saving change** check box

Each sender and receiver PLC is connected to the same external NTP server.

You may configure two redundant NTP servers. If the connection with the primary NTP server is not correct, the 140 NOE 771 11 module is automatically connected to the redundant NTP server. When the connections are properly set, both servers must be synchronized and display the same time value.

⚠ CAUTION

LOSS OF TIME SYNCHRONIZATION

Do not change the NTP server time during operation.

Failure to follow these instructions can result in injury or equipment damage.

NTP Server Time Consistency and System Bits

NTP server time consistency:

- If the NTP server time is consistent with the internal PLC time in `%SW36` to `%SW38` with less than 2 seconds difference, then the time value in `%SW36` to `%SW38` is updated with the last NTP server time received filtered with a slope of 1ms/s.
- If the NTP server time received differs from the internal PLC time in `%SW36` to `%SW38` by more than 2 seconds, then the last NTP server time received is ignored by the PLC, the time value in `%SW36` to `%SW38` is refreshed internally and the bit `%SW39.2` is set to 1 to warn the user.

  In order to have the NTP server time being taken into account by the PLC you can do one of the following actions:

  - ○ reinitialize the application by a cold start
  - ○ download the application
  - ○ restart the PLC
  - ○ set the system bit `%SW39.8` to 1. In this case, the CPU will accept the next NTP server time received without filtering (1ms/s) and without consistency check. After the next NTP server time is received, the `%SW39.8` bit is automatically reset to 0 by the controller.

NOTE: If the system bit `%SW39.8` is set to 1, both sender and receiver PLCs time base can be de synchronized and there is a risk that the safe peer-to-peer communication fails (`S_RD_ETH` DFB `health` output parameter set to 0).

⚠ CAUTION

LOSS OF TIME SYNCHRONIZATION

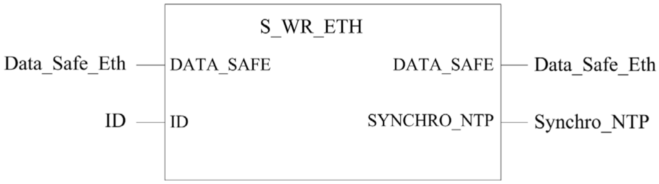
Do not set the system bit `%SW39.8` to 1 continuously.

Failure to follow these instructions can result in injury or equipment damage.

Configuration of S_WR_ETH DFB in the User Program of the Sender PLC

Representation

DFB representation (more details in Unity Pro Safety Block Library):



Description

This DFB calculates data (reserved data containing a CRC and a time stamp) required by the receiver to check and manage errors detected during the safe peer-to-peer communication.

The S_WR_ETH DFB function block has to be called at each cycle in the sender PLC. Within the cycle, it has to be executed in the logic after all required modifications have been performed on the data to be sent. This means that the data to be sent must not be modified by the user within the cycle after the execution of the DFB, otherwise the CRC information used in the reserved data area will not be correct and the safe peer-to-peer communication fails.

You have to assign the ID parameter a unique value that identifies the safe peer-to-peer communication between a sender and a receiver.

⚠ WARNING

LOSS OF ABILITY TO PERFORM SAFETY FUNCTIONS

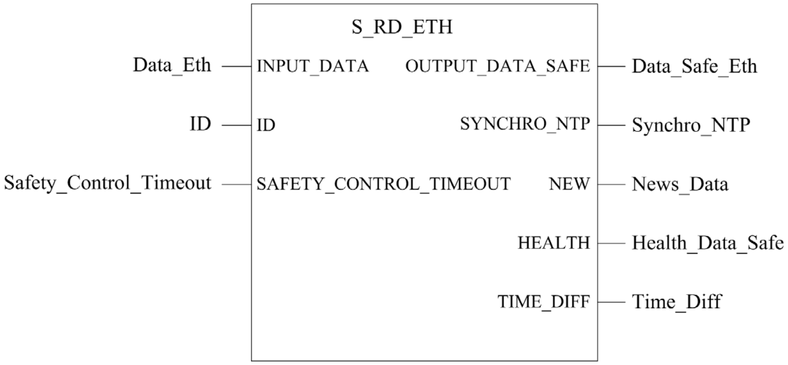
The ID parameter value must be unique and fixed in the network for a sender/receiver pair.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Configuration of S_RD_ETH DFB in the User Program of the Receiver PLC

Representation

DFB representation (more details in Unity Pro Safety Block Library):



Description

This DFB copies the data from the unrestricted memory area to the Safety memory area and guarantees the validity of the received data. The data copy from the unrestricted memory area is not made if the integrity of the data is not correct.

⚠ **WARNING**

LOSS OF ABILITY TO PERFORM SAFETY FUNCTIONS

The S_RD_ETH DFB function block must be called at each cycle in the receiver PLC application and must be executed before the data usage in the cycle.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

The data contained in the output array OUTPUT_DATA_SAFE are considered as safe if, and only if, the output parameter HEALTH is set to 1.

Unity Pro XLS checks that the INPUT_DATA array is allocated to the unrestricted memory area and that the OUTPUT_DATA_SAFE array is allocated to the Safety memory area. If the data arrays are not properly allocated, Unity Pro XLS creates an error message and does not generate the user application code.

HEALTH Bit Description

HEALTH bit meaning:

- = 1, the integrity of the data is correct (CRC) and if the age of the data is lower than the value set in the SAFTETY_CONTROL_TIMEOUT input register.
  The age of the data considered is the time between:
  - the beginning of the cycle where the data are computed in the sender PLC,
  - and the beginning of the cycle where the data are checked in the receiver PLC.
- =0, new valid data are not received in the required time interval (the timer expires and the HEALTH bit is set to 0)
  If the HEALTH bit is set to 0, you must consider the data in the output array OUTPUT_DATA_SAFE as unsafe and react accordingly.

⚠ **WARNING**

LOSS OF ABILITY TO PERFORM SAFETY FUNCTIONS

You must test the HEALTH bit value of the S_RD_ETH DFB at each cycle before using any safe data to manage the safety function.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

ID Parameter Description

You have to assign the ID parameter a unique value that identifies the safe peer-to-peer communication between a sender and a receiver.

⚠ **WARNING**

LOSS OF ABILITY TO PERFORM SAFETY FUNCTIONS

The ID parameter value must be unique and fixed in the network for a sender/receiver pair.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

SYNCHRO_NTP Parameter Description

SYNCHRO_NTP is a logical AND of the following bits:

- %SW39.0
- %SW39.1
- %SW39.2

SYNCHRO_NTP bit meaning:

- =1, the synchronization between sender PLC and receiver PLC is healthy
- =0, the synchronization between sender PLC and receiver PLC is not guaranteed and there is a risk that the safe peer-to-peer communication becomes unhealthy due to internal time slippage (HEALTH bit is set to 0).

You must identify as soon as possible the root cause of the NTP synchronization issue and fix it. The system word `%SW39` helps you to diagnose and fix the issue.

SAFETY_CONTROL_TIMEOUT Parameter Description

The `SAFETY_CONTROL_TIMEOUT` parameter defines the maximum expectation age accepted for data received in the receiver PLC.

`SAFETY_CONTROL_TIMEOUT` parameter value:

- Minimum value: `SAFETY_CONTROL_TIMEOUT` > T1
- Recommended value: `SAFETY_CONTROL_TIMEOUT` > 2 * T1

T1 = $\text{CPU}_{\text{sender}}$ cycle time + Repetitive_rate + Network transmission time + $\text{CPU}_{\text{receiver}}$ cycle time with:

- $\text{CPU}_{\text{sender}}$ cycle time: cycle time of the sender PLC
- Repetitive_rate: time rate for the IO scanner write query from the sender PLC to the receiver PLC
- Network transmission time: time consumed on the Ethernet network for the data transmission from the sender PLC to the receiver PLC
- $\text{CPU}_{\text{receiver}}$ cycle time: cycle time of the receiver PLC

It is very important to note that the value defined for the `SAFETY_CONTROL_TIMEOUT` parameter has a direct effect on the robustness and availability of the safe peer-to-peer communication. If the `SAFETY_CONTROL_TIMEOUT` parameter value is highly greater than T1, the communication will be tolerant to various delays (for example network delays) or corruption errors during the data transmission.

You are responsible for ensuring that the Ethernet network has a load that does not lead to an abnormal delay on the network during data transmission which could lead to the expiration of the timeout. In order to prevent your safe peer-to-peer communication from any abnormal delays due to other non-safety data transmitted on the same network, you can use a dedicated Ethernet network for the safe peer-to-peer protocol.

When commissioning your project, you have to estimate the safe peer-to-peer communication performance by checking the values provided in the output parameter `TIME_DIFF` and evaluating the margin using the value defined in the `SAFETY_CONTROL_TIMEOUT` parameter.

`SAFETY_CONTROL_TIMEOUT` parameter in a Hot Standby system:

- If a Hot Standby system is used in the PLC-PLC safe communication (either as a sender or a receiver), the `SAFETY_CONTROL_TIMEOUT` parameter value has to fit the following additional condition:
  `SAFETY_CONTROL_TIMEOUT` > T1 + 1000 ms + Repetitive_rate + Max($\text{CPU}_{\text{sender}}$ cycle time;$\text{CPU}_{\text{receiver}}$ cycle time)

Configuration of IO Scanning Service

Description

The IO scanning service for safe peer-to-peer communication is used for data transportation:

- from `DATA_SAFE` array (`S_WR_ETH` output parameter) in the Safety memory of the sender PLC
- to the `INPUT_DATA` array (`S_RD_ETH` input parameter) in the unrestricted memory of the receiver PLC

The IO scanning service is configured in Unity Pro XLS *(see Modicon Quantum with Unity, Ethernet Network Modules, User Manual)*, in the Ethernet network configuration of the sender PLC.

The Ethernet network used for IO scanning can be either connected to the Ethernet port of the CPU or to an Ethernet module (140 NOE 771 11).

Configuration

In Unity Pro XLS, configure the IO scanning for safe peer-to-peer communication of the sender PLC respecting the following requirements:

- Configure the data to send in 1 block with a write request.
- Set the **Health Timeout (ms)** parameter value to 300. If the communication conditions do not allow this value, set this parameter to the minimum value allowed by the conditions of the communication.
- Set the **WR length** parameter to 100 (data size is fixed to 100 words).
- Set a value in **WR Master Object** parameter that fit to the address of `DATA_SAFE` output parameter (`S_WR_ETH` DFB) on sender PLC program.
- Set a value in **WR Ref Slave** parameter that fit to the address of `INPUT_DATA` input parameter (`S_RD_ETH` DFB) on receiver PLC program.
- In the receiver PLC, the memory area where the data will be written has to be located in the unrestricted memory area (see parameter **WR Ref Slave**)
- Choose a **Repetitive rate (ms)** value equal to the receiver PLC cycle time.

Safe Peer-to-peer Communication Impacts

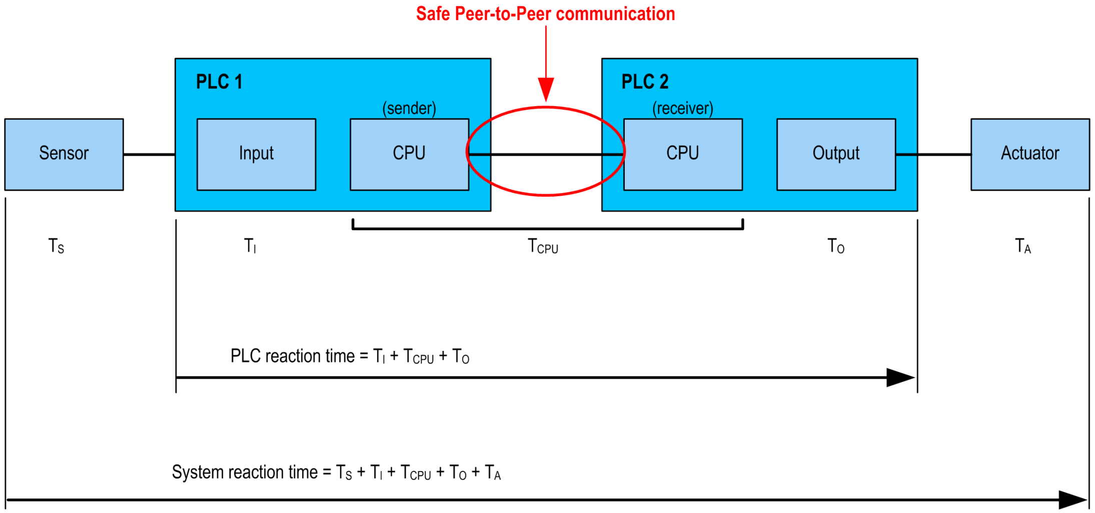
Impact on the CPU Reaction Time

When safe peer-to-peer communication is used to perform the Safety function, the CPU reaction time is directly impacted. The CPU reaction time *(see page 76)* is extended as follows:

CPU reaction time = (1 + N_{CRC}) x CPU_{sender} cycle time + $CPU_{receiver}$ cycle time + `SAFETY_CONTROL_TIMEOUT`

This equation has to be used to calculate the system reaction time.

The following figure shows the safety system reaction time:


Safe Peer-to-Peer communication
PLC 1
(sender)
PLC 2
(receiver)
Sensor
Input
CPU
CPU
Output
Actuator
T_S
T_I
T_{CPU}
T_O
T_A
PLC reaction time = $T_I + T_{CPU} + T_O$
System reaction time = $T_S + T_I + T_{CPU} + T_O + T_A$

Impact on the Maximum CPU Cycle Time

When safe peer-to-peer communication is used to perform the Safety function, in order to ensure that the system reaction time is smaller than the process Safety time, the maximum CPU cycle time *(see page 76)* becomes:

((1 + N_{CRC}) x Max. CPU_{sender} cycle time + Max. $CPU_{receiver}$ cycle time) < (PST - T_I - T_O - T_S -T_A - `SAFETY_CONTROL_TIMEOUT`)

Impact on PFD/PFH Calculation

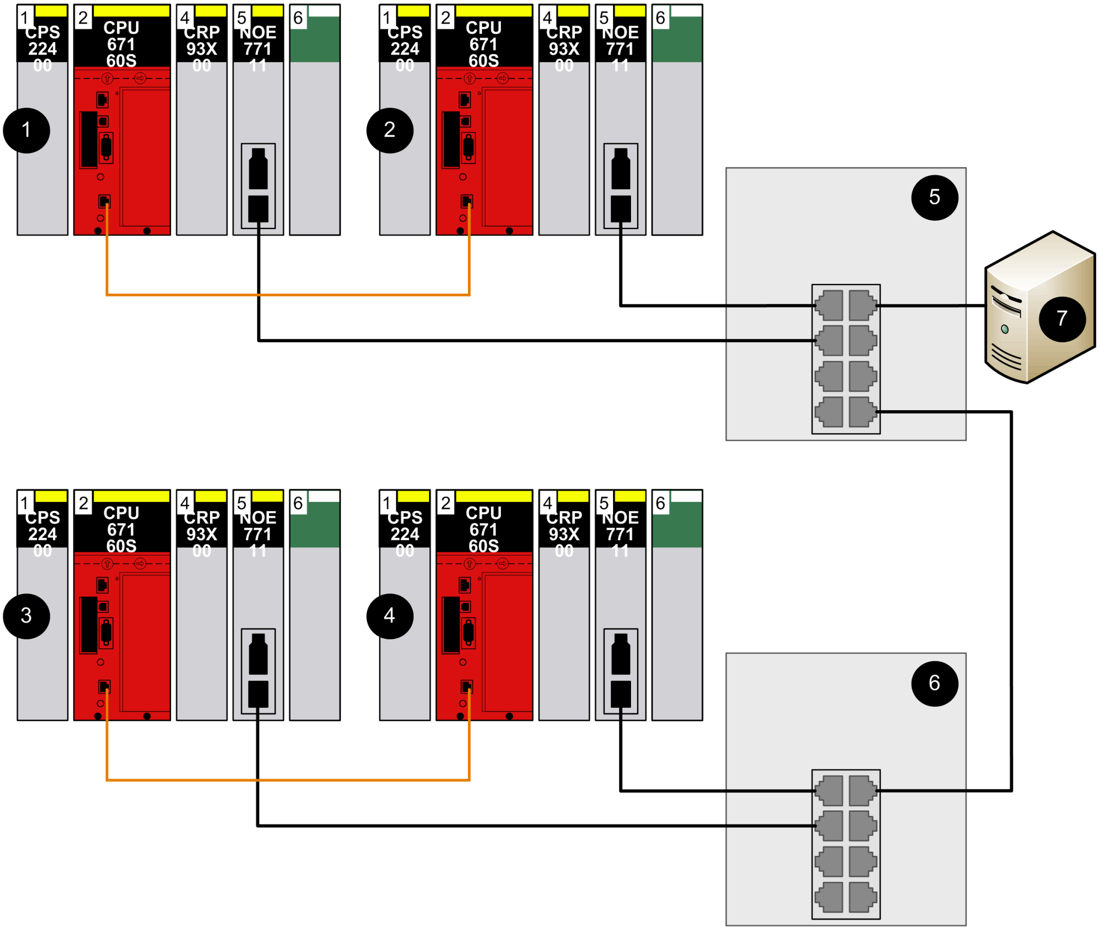
When using a safe peer-to-peer communication in the Safety loop, you have to include the contribution of the sender PLC and the receiver PLC in the calculation of the PFD/PFH values of the system.

On the contrary, all modules on the Ethernet network which are part of the black channel like 140 NOE 771 11 modules, switches, NTP servers do not contribute to the calculation of the PFD/PFH values of the system.

Example of Configuration, Parameters and Performance Results

Configuration Example

In the following Ethernet network design, 2 Hot Standby configurations (based on 140 CPU 671 60 S CPU) use a safe peer-to-peer communication to exchange data:



1 Sender PLC Primary local rack
2 Sender PLC Standby local rack
3 Receiver PLC Primary local rack
4 Receiver PLC Standby local rack
5 499NES17100 switch
6 499NES17100 switch
7 NTP server

NOTE: The Ethernet RIO network is not represented in the previous example.

Parameters and Performance Results

The parameters used in the previous example are as follows:

- Sender PLC:
  - size of words sent by IO scanning to the receiver PLC = 100 words (safe data only)
  - no messaging configured
  - no global data configured
- Receiver PLC:
  - no messaging configured
  - no global data configured
  - no IO scanning configured

The measurements done in the previous example are as follows:

Parameters			Maximum TIME_DIFF value observed on the output parameter of S_RD_ETH DFB in normal operation (in ms after 8 hours of operation)	Maximum TIME_DIFF value observed on the output parameter of S_RD_ETH DFB after a Switchover between Primary and Standby on the sender PLC (in ms after 100 Switchover performed)
Sender PLC Cycle Time (ms)	Repetitive Rate (ms)	Receiver PLC Cycle Time (ms)		
60	100	100	223	884
100	100	100	296	760
150	150	150	446	832
200	200	200	608	953

Section 4.5
PLC-HMI Communication

PLC-HMI Communication Description

Introduction

A HMI is allowed to read data from a Safety PLC. However, it is only allowed to write to the unrestricted memory area of the PLC, see also *Memory Area Description, page 104*. The Quantum Safety PLC is able to communicate with HMIs using the following:

- Modbus TCP (either with CPU or NOE module)
- Modbus Plus
- Modbus RS232 / RS485

The communication between PLC and HMI is not configured in Unity Pro XLS. Therefore, it cannot be controlled by it and the Quantum Safety CPU protects itself against writing from a HMI.

Write Protection Description

The Safety memory area of the Safety PLC is write protected and you are not allowed to write to it. If you do not obey this rule, the PLC does not execute your write command, see also *Write Protection Description, page 104*.

Writing in Maintenance Mode

Even in Maintenance Mode, there is a write protection of the Safety memory area for other PLCs and HMIs. But with Unity Pro XLS, you are able to modify and tune data.

With Unity Pro XLS, it is possible to

- modify logic.
- set values.
- force values.
- debug.

By using the Schneider Electric OPC server OFS or the web server of the PLC, it is also possible to modify data in the Safety memory area when in Maintenance Mode.

⚠ WARNING

RISK OF PROCESSING FORCED DATA

Follow the latest version of the TÜV document *Maintenance Override* if you use the Maintenance Mode to modify Safety data. You can find it on the TÜV Rheinland Group website *http://www.tuvasi.com/*.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Chapter 5
Checklists

Introduction

For a system to perform Safety Functions, installing and configuring, programming, commissioning, and operating must meet the Safety requirements of the IEC 61508. To ensure that Safety aspects are observed, Schneider Electric recommends that you use the following checklists. However, these lists are not exhaustive and you are fully responsible for observing all Safety requirements mentioned in the IEC 61508 as well as in this manual.

What Is in This Chapter?

This chapter contains the following topics:

Checklist for Configuring Safety-Related Systems

Introduction

This list is not exhaustive and you are fully responsible for observing all Safety requirements mentioned in the IEC 61508 as well as in this manual.

Checklist

Schneider Electric recommends that you use the following checklist for configuring your Safety-Related System:

Checks	Reference in this Manual	Done	Remarks
Check and verify the PFD/PFH values of the complete Safety loop.	*Functional Safety Certification, page 16*	□	
Respect all rules described in the following reference manuals: ● *Quantum with Unity Pro Hardware Reference Manual* ● *Quantum with Unity Pro Discrete and Analog I/O Reference Manual* ● *Grounding and Electromagnetic Compatibility of PLC Systems User Manual* ● *Modicon Remote I/O Cable System Planning and Installation Guide* ● *Modicon Quantum Hot Standby with Unity User Manual* ● *Premium, Atrium and Quantum using Unity Pro Communication services and architectures Reference manual* ● *Quantum TCPIP/IP Configuration User Manual* ● *Modicon Quantum with Unity Ethernet Network Modules User Manual*		□	
Test and verify the complete configuration and wiring as part of the commissioning.		□	
Use certified Safety and non-interfering modules only.	*Functional Safety Certification, page 16*	□	
Use modules with the certified firmware versions only. It is possible to check the firmware version of the CPU, CRP/CRA and NOE modules as well as that of the CPU Ethernet and CPU Hot Standby processors, with the OSloader. The firmware of the Safety I/O modules is displayed on the label on the housing.	*Functional Safety Certification, page 16*	□	

Checks	Reference in this Manual	Done	Remarks
Configure the maximum scan time correctly and appropriately to the process.	*Requirements for Monitoring, page 74*, *Safety Mode Restrictions, page 90*, *Process Safety Time, page 75*	□	
Use an application password to protect your SIL3 application against unauthorized access.	*Application Password, page 86*	□	
Use only high availability RIO modules (140 CRP 932 00 and 140 CRA 932 00), which provide dual cabling.	*Description of the RIO Adapters, page 55*, *Description of the CPU-I/O Communication, page 39*	□	
Protect the process power supply of the digital output modules by an appropriate fuse.	*Wiring Information, page 51*	□	
Use 2 power supply modules per rack and drop to improve the availability of your system. Mounting the 2 power supply modules on each end of the rack or drop provides better heat dissipation.	*Power Supply for the Quantum Safety PLC, page 54*	□	
Check in each drop that 1 power supply module is able to deliver the complete power consumption.	*Power Supply for the Quantum Safety PLC, page 54*	□	
Make sure that the addresses of all CRA modules are configured correctly.	*Description of the RIO Adapters, page 55*	□	
Do not write data to the Safety memory area from other devices (PLCs, HMI, and so on).	*Communication, page 103*	□	
Do not download the Ethernet processor firmware while the PLC is running in Safety Mode.	*Safety Mode Restrictions, page 90*	□	

Checklist for Programming SIL3 Applications

Introduction

This list is not exhaustive and you are fully responsible for observing all Safety requirements mentioned in the IEC 61508 as well as in this manual.

Checklist

Schneider Electric recommends that you use the following checklist for programming your SIL3 application:

Checks	Reference in this Manual	Done	Remarks
Check the consistency of Unity Pro XLS regularly.	*Checking the Programming Environment, page 96*	□	
Check the correctness of your project.	*Programming Requirements, page 29*	□	
Test and verify the complete logic as part of the commissioning.		□	
Configure the maximum %M and %MW correctly.	*Memory Area Description, page 104*	□	
Configure the maximum unrestricted areas for %M and %MW correctly.	*Description of the Maximum CPU Cycle Time, page 76*	□	
Check that the configured maximum UMA for %M and %MW is downloaded correctly (check with %SW110 and %SW111).	*Memory Area Description, page 104*	□	
Check the correct usage of non-Safety-related data from the unresticted memory area with `S_SMOVE_***` function blocks.	*Memory Area Description, page 104*	□	
Check the range of WORD data of non-Safety-related data from the unresticted memory area by configuring the `S_SMOVE_WORD` function block.	*Memory Area Description, page 104*	□	
Do not use use conditional execution of Safety logic sections.	*Requirements for the Program Structure, page 72*	□	

Checks	Reference in this Manual	Done	Remarks
Do not use jumps to labels inside FBD and LD logic.	*Requirements for Language Elements, page 72*	□	
Program the non-Safety logic for non-interfering I/Os in separate sections.	*Available Non-Interfering Products, page 19*	□	
Indicate the non-Safety-Related variables with an appropriate naming convention and comment.	*Memory Area Description, page 104*	□	
Make sure that inputs or outputs of non-interfering I/O modules are not used for calculating Safety-Related outputs.	*Description of the I/O and Time Stamp Modules, page 56*	□	
Do not monitor concurrently a huge amount of data in Unity Pro XLS (leads to increase of scan time).	*Requirements for Monitoring, page 74*	□	
In the project settings, switch on all options for warnings during analysis. Check all warnings and make sure that they are not critical and that the behavior is intended.	*Checks for Programming, page 73*	□	

Checklist for I/O Modules

Introduction

This list is not exhaustive and you are fully responsible for observing all Safety requirements mentioned in the IEC 61508 as well as in this manual.

Checklist for the I/O Modules

Schneider Electric recommends that you use the following checklist for your I/O modules:

Checks	Reference in this Manual	Done	Remarks
Do not use Ethernet I/Os.	restrictions on I/Os *(see page 39)*	□	
Do not use Modbus Plus I/Os.	restrictions on I/Os *(see page 39)*	□	
Do not use non-interfering I/Os for Safety-Related functions.	descr. of I/O modules *(see page 56)*	□	
The wiring of the digital inputs must be de-energized to trip (a wiring fault must be equivalent to the de-energized state).	wiring of SDI *(see page 48)*	□	
Use appropriate grounding equipment for the analog input shielded wires.	wiring of SAI *(see page 45)*	□	
In burner management applications, the analog inputs must be monitored for grounding faults (leakage of current).	special req. for appl. standard *(see page 139)*	□	
Check that the configured timeout state of the output modules is appropriate for the connected device and the controlled process.	description of the timeout state *(see page 52)*	□	
In a redundant I/O system, use the 2 I/O channels on separate modules which should be located in separate drops.	redundant I/O configuration *(see page 64)*	□	
Use an appropriate wire type/size to connect the inputs/outputs of the I/O modules with the sensors and actuators.	RIO adapter *(see page 55)*	□	
For unused inputs of the Safety analog input module, the health bit of unused inputs should be masked in the health word of the module in your application logic.	wiring of SAI *(see page 45)*	□	
Check that sensors and actuators connected to the I/O modules respect the specified values and limits of the I/O modules.	process safety time *(see page 75)*	□	

Checks	Reference in this Manual	Done	Remarks
Use the red labels for the terminal blocks provided with the Safety I/O modules to indicate clearly the Safety modules.	gen. inf. on safety I/Os *(see page 39)*	□	

Checklist for Configuring Safe Peer-to-Peer Communication

Introduction

This list is not exhaustive and you are fully responsible for observing all Safety requirements mentioned in the IEC 61508 as well as in this manual.

Checklist for the I/O Modules

Schneider Electric recommends that you use the following checklist for configuring your safe peer-to-peer communication:

Checks	Reference in this Manual	Done	Remarks
Check and verify the PFD/PFH values of the complete Safety loop by taking into account the safe peer-to-peer communication: both sender CPU and receiver CPU contribute to the calculation.	Impact on PFD/PFH Calculation *(see page 120)*	□	
Respect all rules described in the following reference manuals: ● *Grounding and Electromagnetic Compatibility of PLC Systems User Manual* ● *Modicon Quantum Hot Standby with Unity User Manual* ● *Quantum TCPIP/IP Configuration User Manual* ● *Modicon Quantum with Unity Ethernet Network Modules User Manual*	Related Documents *(see page 9)*	□	
Verify that the NTP service is configured on each receiver PLC and sender PLC by using the non interfering 140 NOE 771 11 module in Unity Pro XLS.	Configuration of NTP Service *(see page 113)*	□	
Verify that each receiver PLC and sender PLC is connected to a same external NTP server and have the same **Time Zone** parameter configured.	Configuration of NTP Service *(see page 113)*	□	
In Unity Pro XLS, for each receiver PLC and sender PLC, make sure that NTP configuration parameters are as follows: ● **Automatically adjust clock for daylight saving change** check box is unchecked ● **Polling period** value is set to 20s	Configuration of NTP Service *(see page 113)*	□	

Checks	Reference in this Manual	Done	Remarks
Check that the sender PLC and receiver PLC application do not set the %SW39.8 system bit to 1 continuously.	NTP Server Time Consistency and System Bits *(see page 114)*	□	
On the sender PLC, verify that the S_WR_ETH DFB function block is called at each cycle and executed in the logic after all required modifications have been performed on the data to be sent.	Configuration of S_WR_ETH DFB in the User Program of the Sender PLC *(see page 115)*	□	
On the receiver PLC, verify that the S_RD_ETH DFB function block is called at each cycle and executed before the data usage in the cycle.	Configuration of S_RD_ETH DFB in the User Program of the Receiver PLC *(see page 116)*	□	
Check that the ID parameter value of the S_RD_ETH and S_WR_ETH function blocks are identical in the sender PLC and the receiver PLC and that the value is unique and fixed in the complete system for a sender/receiver pair.	Configuration of S_WR_ETH DFB in the User Program of the Sender PLC *(see page 115)*	□	
Check that the receiver PLC program monitors the HEALTH output parameter of the S_RD_ETH function block. If the HEALTH parameter is set to 0, the data in the OUTPUT_DATA_SAFE array are considered as unsafe in the Safety loop and the receiver PLC must react accordingly.	Configuration of S_RD_ETH DFB in the User Program of the Receiver PLC *(see page 117)*	□	

Checks	Reference in this Manual	Done	Remarks
On IO Scanning configuration of the sender PLC, check that: - The data are sent in 1 block with a write request to the receiver PLC. - The **Health Timeout (ms)** parameter value is equal to 300 ms or to the minimum value allowed by the conditions of the communication. - The **WR length** parameter is set to 100. - The value in **WR Master Object** parameter fits to the source address of `DATA_SAFE` output parameter (`S_WR_ETH` DFB) on sender PLC program. - The value in **WR Ref Slave** parameter fits to the address of `INPUT_DATA` input parameter (`S_RD_ETH` DFB) on receiver PLC program. - In the receiver PLC, the memory area where the data will be written is located in the unrestricted memory area. - The **Repetitive rate (ms)** value is equal to the receiver PLC cycle time.	Configuration of the IO Scanning Service *(see page 119)*	□	
If safe peer-to-peer communication is used, monitor the status of the NTP synchronization (`%SW39` or `SYNCHRO_NTP` output parameter of the `S_RD_ETH` and `S_WR_ETH` function blocks) and signal a detected fault to the maintenance personnel.	NTP Server Time Consistency and System Bits *(see page 114)*	□	
If safe peer-to-peer communication is used, monitor the status of the `HEALTH` output parameter of the `S_RD_ETH` function block.	Configuration of S_RD_ETH DFB in the User Program of the Receiver PLC *(see page 117)*	□	

Checklist for Operation, Maintenance, and Repair

Introduction

This list is not exhaustive and you are fully responsible for observing all Safety requirements mentioned in the IEC 61508 as well as in this manual.

Checklist

Schneider Electric recommends that you use the following checklist for operation, maintenance, and repair of your Safety-Related System:

Checks	Reference in this Manual	Done	Remarks
Define a standard operating procedure (SOP) for operation, maintenance, and repair of the Safety instrumented system and ensure that it is respected.		□	
Define a maintenance plan for your Safety-Related System according to the proof test interval.	*Proof Test Interval, page 22*	□	
Maintain your Safety-Related System according to your maintenance plan.		□	
Create backups of your SIL3 project on a regular basis.	*Project Backups, page 100*	□	
When changing the Safety-Related System, follow the rules of the IEC61508-1, chapters 7.15 and 7.16 (even if only non-Safety-Related parts are modified).		□	
Follow the guidelines of the *Maintenance Override* TÜV document when using forcing (available on *http://www.tuvasi.com/*).	*Forcing, page 94*	□	
Check that forcing is switched off after the maintenance operation (either as part of the application or by an appropriate standard operating procedure).	*Forcing, page 94*	□	
Monitor the status of the Safety I/O modules (health, out of range, overload, invalid channel), see also the *Quantum with Unity Pro Discrete and Analog I/O Reference Manual*.	*Description of the RIO Adapters, page 55*, *Description of the CPU-I/O Communication, page 39*	□	

Checks	Reference in this Manual	Done	Remarks
In a redundant I/O system, signal a fault in 1 of the redundant modules to the maintenance personnel.	*Safety I/O Modules in High Availability Configurations, page 40*	□	
Signal a fault in 1 cable of the dual cable remote I/O system to the maintenance personnel.	*Description of a Safety Hot Standby Configuration, page 35*, *Description of the RIO Adapters, page 55*, *Description of the CPU-I/O Communication, page 39*	□	
In a redundant power supply configuration, signal a fault of 1 of the 2 power supply modules to the maintenance personnel.	*Power Supply for the Quantum Safety PLC, page 54*	□	
In a HSBY system, use the `S_HSBY_SWAP` function block regularly (for example once a week) to check the ability of the standby controller to take over.	*Availability of the Hot Standby Functions, page 36*	□	
When replacing a CRA module, make sure that the address is configured correctly.	*Description of the RIO Adapters, page 55*	□	
Make sure that your personnel possess all information and skills required to install, run, and maintain the Safety-Related System correctly.	*Training, page 28*	□	
Make sure to follow the specified operating conditions regarding EMC, electrical, mechanical, and climatic influences.	*Hardware Requirements, page 29*	□	

⚠ DANGER

RISK OF LOSING THE SAFETY FUNCTION DURING COMMISSIONING AND MAINTENANCE

All modifications of the running system must follow the requirements of the IEC 61508.

Failure to follow these instructions will result in death or serious injury.

Chapter 6
Special Requirements for Application Standards

Special Requirements for Application Standards

Fire and Gas Systems

Fire and gas systems should be integrated in accordance with EN 54.

Fire and gas applications must operate continuously to provide protection. As a result, the following industry guidelines apply:

- If inputs and outputs are energized to mitigate a problem, the PLC system must detect open and short circuits in the wiring between the PLC and the field devices and must raise alarms.
- The entire PLC system must have redundant power supplies. Further, the power supplies that are required to activate critical outputs and to read Safety-critical inputs must be redundant. All power supplies must be monitored for proper operation.
- De-energized outputs may be used for normal operation. To initiate the actions to mitigate a problem, the outputs are energized. This type of system shall monitor the critical output circuits to help ensure that they are properly connected to the end devices.

In fire and gas applications, the Safety analog input modules must be monitored for ground faults (leakage of current). The wires should be connected potential-free. With a shunt resistor (for instance 250 Ω) between the ground rail of the grounding kit and the earth ground, a voltage can be measured in case of a leakage of the current on 1 of the analog inputs. This voltage must be supervised to detect a leakage.

Emergency Shutdown Systems

In Emergency Shutdown systems, the Safe state of the plant is a de-energized or low (0) state.

Burner Management Systems

In burner management systems, the Safe state of the plant is a de-energized or low (0) state.

If a Safety-Related System is required to conform with the EN 298 standard for automatic gas burner control systems, the PLC throughput time should ensure that a Safe shutdown can be performed within 1 second after a problem in the process is detected. For the calculation, see *Process Safety Time, page 75*.

A stabilized power supply of 20 VDC to 25 VDC must be used for the field power.

Appendices

Introduction

The appendices contain information on the IEC 61508 and its SIL policy. Further, technical data of the Safety and non-interfering modules are provided and example calculations are carried out.

What Is in This Appendix?

The appendix contains the following chapters:

Appendix A
IEC 61508

Introduction

This chapter provides information on the Safety concepts of the IEC 61508 in general and its SIL policy in particular.

What Is in This Chapter?

This chapter contains the following topics:

General Information on the IEC 61508

Introduction

Safety-Related Systems are developed for use in processes in which risks to humans, environment, equipment and production must be kept at an acceptable level. The risk depends on the severity and likelihood, thereby defining the necessary measures of protection.

Concerning the Safety of processes, there are 2 sides to be considered:

- the regulations and requirements defined by official authorities in order to protect humans, environment, equipment, and production
- the measures by which these regulations and requirements are fulfilled

IEC 61508 Description

The technical standard defining the requirements for Safety-Related Systems is

- the IEC 61508.

It deals with the Functional Safety of electrical, electronic or programmable electronic Safety-Related Systems. A Safety-Related System is a system that is required to perform 1 or more specific functions to ensure risks are kept at an acceptable level. Such functions are defined as Safety Functions. A system is defined functionally Safe if random, systematic, and common cause failures do not lead to malfunctioning of the system and do not result in injury or death of humans, spills to the environment and loss of equipment and production.

The standard defines a generic approach to all lifecycle activities for systems that are used to perform Safety Functions. It constitutes procedures to be used for the design, the development, and the validation of both hardware and software applied in Safety-Related Systems. Further, it determines rules concerning both the management of Functional Safety and documentation.

IEC 61511 Description

The Functional Safety requirements defined in the IEC 61508 are refined specifically for the process industry sector in the following technical standard:

- the IEC 61511: Functional safety - safety instrumented systems for the process industry sector

This standard guides the user in the application of a Safety-Related System, starting from the earliest phase of a project, continuing through the start up, covering modifications and eventual decommissioning activities. In summary, it deals with the Safety Lifecycle of all components of a Safety-Related System used in the process industry.

Risk Description

The IEC 61508 is based on the concepts of risk analysis and Safety Function. The risk depends on severity and probability. It can be reduced to a tolerable level by applying a Safety Function that consists of an electrical, electronic or programmable electronic system. Further, it should be reduced to a level that is as low as reasonably practicable.

In summary, the IEC 61508 views risks as follows:

- Zero risk can never be reached.
- Safety must be considered from the beginning.
- Intolerable risks must be reduced.

SIL Policy

Introduction

The SIL value evaluates the robustness of an application against failures, thus indicating the ability of a system to perform a Safety Function within a defined probability. The IEC 61508 specifies 4 levels of Safety performance depending on the risk or impacts caused by the process for which the Safety-Related System is used. The more dangerous the possible impacts are on community and environment, the higher the Safety requirements are to lower the risk.

SIL Value Description

Discrete level (1 out of a possible 4) for specifying the Safety Integrity requirements of the Safety Functions to be allocated to the Safety-Related Systems, where Safety Integrity Level 4 has the highest level of Safety Integrity and Safety Integrity Level 1 has the lowest, see *SILs for Low Demand, page 147*.

SIL Requirements Description

To achieve Functional Safety, 2 types of requirements are necessary:

- Safety Function requirements, defining what Safety Functions have to be performed
- Safety Integrity requirements, defining what degree of certainty is necessary that the Safety Functions are performed

The Safety Function requirements are derived from hazard analysis and the Safety Integrity ones from risk assessment.

They consist of the following quantities:

- Mean time between failures
- Probabilities of failure
- Failure rates
- Diagnostic coverage
- Safe failure fraction
- Hardware fault tolerance

Depending on the level of Safety Integrity, these quantities must range between defined limits.

SIL Rating Description

As defined in the IEC 61508, the SIL value is limited by both the Safe Failure Fraction (SFF) and the hardware fault tolerance (HFT) of the subsystem that performs the Safety Function. A HFT of n means that n+1 faults could cause a loss of the Safety Function, the Safe state cannot be entered. The SFF depends on failure rates and diagnostic coverage.

The following table shows the relation between SFF, HFT, and SIL for complex Safety-Related subsystems according to IEC 61508-2, in which the failure modes of all components cannot be completely defined:

SFF	HFT=0	HFT=1	HFT=2
SFF ≤ 60%	-	SIL1	SIL2
60% < SFF ≤ 90%	SIL1	SIL2	SIL3
90% < SFF ≤ 99%	SIL2	SIL3	SIL4
SFF > 99%	SIL3	SIL4	SIL4

There are 2 ways to reach a certain Safety Integrity Level:

- via increasing the HFT by providing additional independent shutdown paths
- via increasing the SFF by additional diagnostics

SIL-Demand Relation Description

The IEC 61508 distinguishes between low demand mode and high demand (or continuous) mode of operation.

In low demand mode, the frequency of demand for operation made on a Safety-Related System is not greater than 1 per year and not greater than twice the proof test frequency. The SIL value for a low demand Safety-Related System is related directly to its average probability of failure to perform its Safety Function on demand or, simply, probability of failure on demand (PFD).

In high demand or continuous mode, the frequency of demand for operation made on a Safety-Related System is greater than 1 per year and greater than twice the proof test frequency. The SIL value for a high demand Safety-Related System is related directly to its probability of a dangerous failure occurring per hour or, simply, probability of failure per hour (PFH).

SILs for Low Demand

The following table lists the requirements for a system in low demand mode of operation:

Safety Integrity Level	Probability of Failure on Demand
4	≥ 10^{-5} to < 10^{-4}
3	≥ 10^{-4} to < 10^{-3}
2	≥ 10^{-3} to < 10^{-2}
1	≥ 10^{-2} to < 10^{-1}

SILs for High Demand

The following table lists the requirements for a system in high demand mode of operation:

Safety Integrity Level	Probability of Failure per Hour
4	$\geq 10^{-9}$ to $< 10^{-8}$
3	$\geq 10^{-8}$ to $< 10^{-7}$
2	$\geq 10^{-7}$ to $< 10^{-6}$
1	$\geq 10^{-6}$ to $< 10^{-5}$

For SIL3, the required probabilities of failure for the complete Safety integrated system are:

- PFD $\geq 10^{-4}$ to $< 10^{-3}$ for low demand
- PFH $\geq 10^{-8}$ to $< 10^{-7}$ for high demand

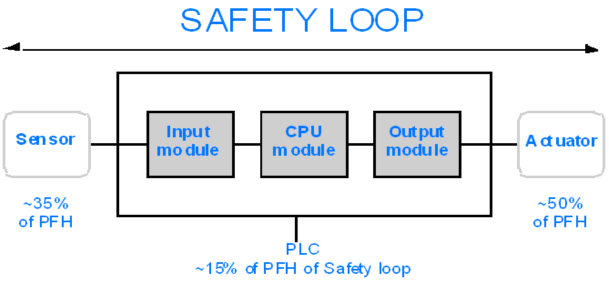
Safety Loop Description

The Safety loop to which the Quantum Safety PLC consists of the following 3 parts:

- Sensors
- Quantum Safety PLC with Safety CPU and Safety I/O modules
- Actuators

A backplane or a remote connection with CRA/CRP do not destroy a Safety Loop. Backplanes, CRP and CRA modules are part of a “black channel”. This means that the data exchanged by I/O and PLC cannot be corrupted without detection by the receiver.

The following figure shows a typical Safety loop:



As shown in the figure above, the contribution of the PLC is only 10-20% because the probability of failure of sensors and actuators is usually quite high.

A conservative assumption of 10% for the Safety PLC’s contribution to the overall probability leaves more margin for the user and results in the following required probabilities of failure for the Safety PLC:

- PFD $\geq 10^{-5}$ to $< 10^{-4}$ for low demand
- PFH $\geq 10^{-9}$ to $< 10^{-8}$ for high demand

PFD Equation Description

The IEC 61508 assumes that half of the failures end in a Safe state. Therefore, the failure rate λ is divided into

- λ_S - the safe failure and
- λ_D - the dangerous failure, itself composed of
  - λ_{DD} - dangerous failure detected by the internal diagnostic
  - λ_{DU} - dangerous failure undetected.

The failure rate can be calculated by using the mean time between failures (MTBF), a module specific value, as follows:

$\lambda = 1/MTBF$

The equation for calculating the probability of failure on demand is:

$PFD(t) = \lambda_{DU} \times t$

t represents the time between 2 proof tests.

The probability of failure per hour implies a time interval of 1 hour. Therefore, the PFD equation is reduced to the following one:

$PFH = \lambda_{DU}$

Appendix B
System Objects

Introduction

This chapter describes the system bits and words of the Quantum Safety PLC.

Note: The symbols associated with each bit object or system word mentioned in the descriptive tables of these objects are not implemented as standard in the software, but can be entered using the data editor.

It is suggested that the symbol names associated with the system bits and system words that appear on the following pages be implemented to provide continuity and ease of understanding. Example: `%S0` COLDSTART (the user can select another word to replace COLDSTART).

What Is in This Chapter?

This chapter contains the following sections:

Section B.1
System Bits

Introduction

This section describes the system bits of the Quantum Safety PLC.

For your convenience, all system bits of standard Quantum PLCs are listed but only explained further if used in the Quantum Safety PLC.

What Is in This Section?

This section contains the following topics:

System Bit Introduction

General

The Quantum PLCs use %Si system bits which indicate the state of the PLC, or they can be used to control how it operates.

These bits can be tested in the user program to detect any functional development.

Some of these bits must be reset to their initial or normal state by either the program or the user. Other bits are automatically reset by the system. Finally, there are bits which only display the status of the PLC.

Description of the System Bits %S0 to %S13

Detailed Description

NOTE: Not all of the system bits can be used in the Quantum Safety PLC. The unusable system bits are marked in the **Quant. Safety** column with no.

The following table gives a description of the system bits %S0 to %S13:

Bit **Symbol**	Function	Description	Initial State	Write Access	Quant. Safety
%S0 COLDSTART	cold start	Normally at 0, this bit is set to 1 by: ● power restoration with loss of data (battery related), ● the user program, ● the terminal, ● a change of cartridge, This bit is set to 1 during the first complete restored cycle of the PLC either in RUN or in STOP mode. It is reset to 0 by the system before the following cycle. %S0 is not always set in the first scan of the PLC. If a signal set for every start of the PLC is needed, %S21 should be used instead.	1 (1 cycle)	no	yes
%S1 WARMSTART	warm restart	see chapter "System Bits" in the *Unity Pro Program Languages and Structure Reference Manual*	0	no	no
%S4 TB10MS	time base 10 ms	see chapter "System Bits" in the *Unity Pro Program Languages and Structure Reference Manual*	-	no	no
%S5 TB100MS	time base 100 ms	see chapter "System Bits" in the *Unity Pro Program Languages and Structure Reference Manual*	-	no	no
%S6 TB1SEC	time base 1 s	see chapter "System Bits" in the *Unity Pro Program Languages and Structure Reference Manual*	-	no	no
%S7 TB1MIN	time base 1 min	see chapter "System Bits" in the *Unity Pro Program Languages and Structure Reference Manual*	-	no	no
%S10 IOERR	input/output fault	Normally at 1, this is set to 0 when an I/O fault on an in-rack module or device on Fipio is detected (e.g. non-compliant configuration, exchange fault, hardware fault, etc.). The %S10 bit is reset to 1 by the system as soon as the fault disappears.	1	no	yes
%S11 WDG	watchdog overflow	Normally at 0, this is set to 1 by the system as soon as the task execution time becomes greater than the maximum execution time (i.e. the watchdog) declared in the task properties.	0	no	yes

Bit Symbol	Function	Description	Initial State	Write Access	Quant. Safety
%S12 PLCRUNNING	PLC in RUN	This bit is set to 1 by the system when the PLC is in RUN. It is set to 0 by the system as soon as the PLC is no longer in RUN (STOP, INIT, etc.).	0	no	yes
%S13 1RSTSCANRUN	first cycle after switching to RUN	Normally set to 0, this is set to 1 by the system during the first cycle of the master task after the PLC is set to RUN.	-	no	yes

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

On Quantum Safety PLCs, communication interruptions from NOE, CRA or CRP modules are not reported on bit %S10.

Make certain that these system bits are used correctly.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Description of the System Bits %S15 to %S21

Detailed Description

NOTE: Not all of the system bits can be used in the Quantum Safety PLC. The unusable system bits are marked in the **Quant. Safety** column with no.

The following table gives a description of the system bits %S15 to %S21:

Bit Symbol	Function	Description	Initial State	Write Access	Quant. Safety
%S15 STRINGERROR	character string fault	see chapter "System Bits" in the *Unity Pro Program Languages and Structure Reference Manual*	0	yes	no
%S16 IOERRTSK	task input/output fault	Normally set to 1, this bit is set to 0 by the system when a fault occurs on an in-rack I/O module or a Fipio device configured in the task. This bit must be reset to 1 by the user.	1	yes	yes
%S17 CARRY	rotate or shift output	normally at 0 During a rotate or shift operation, this bit takes the state of the outgoing bit.	0	no	yes
%S18 OVERFLOW	overflow or arithmetic error	Normally set to 0, this bit is set to 1 in the event of a capacity overflow if there is ● a result greater than + 32 767 or less than - 32 768, in single length, ● result greater than + 65 535, in unsigned integer, ● a result greater than + 2 147 483 647 or less than - 2 147 483 648, in double length, ● result greater than +4 294 967 296, in double length or unsigned integer, ● real values outside limits, ● division by 0, ● the root of a negative number, ● forcing to a non-existent step on a drum, ● stacking up of an already full register, emptying of an already empty register. It must be tested by the user program after each operation where there is a risk of overflow, and then reset to 0 by the user if there is indeed an overflow. When the %S18 bit switches to 1, the application stops in error state if the %S78 bit has been set to 1.	0	yes	yes
%S19 OVERRUN	task period overrun (periodical scanning)	Normally set to 0, this bit is set to 1 by the system in the event of a time period overrun (i.e. task execution time is greater than the period defined by the user in the configuration or programmed into the %SW word associated with the task). The user must reset this bit to 0. Each task manages its own %S19 bit.	0	yes	yes

Bit Symbol	Function	Description	Initial State	Write Access	Quant. Safety
%S20 INDEXOVF	Index overflow	Normally set to 0, this bit is set to 1 when the address of the indexed object becomes less than 0 or exceeds the number of objects declared in the configuration. In this case, it is as if the index were equal to 0. It must be tested by the user program after each operation where there is a risk of overflow, and then reset to 0 if there is indeed an overflow. When the %S20 bit switches to 1, the application stops in error state if the %S78 bit has been set to 1.	0	yes	no
%S21 1RSTTASKRUN	first task cycle	Tested in a task (Mast, Fast, Aux0, Aux1, Aux2 Aux3), the bit %S21 indicates the first cycle of this task. %S21 is set to 1 at the start of the cycle and reset to zero at the end of the cycle. **Notes**: The bit %S21 does not have the same meaning in PL7 as in Unity Pro.	0	no	yes

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

On Quantum Safety PLCs, communication interruptions from NOE, CRA or CRP modules are not reported on bit %S16.

Make certain that these system bits are used correctly.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Description of the System Bits %S30 to %S51

Detailed Description

NOTE: Not all of the system bits can be used in the Quantum Safety PLC. The unusable system bits are marked in the **Quant. Safety** column with no.

The following table gives a description of the system bits %S30 to %S51:

Bit Symbol	Function	Description	Initial State	Write Access	Quant. Safety
%S30 MASTACT	activation/deactivation of the master task	see chapter "System Bits" in the *Unity Pro Program Languages and Structure Reference Manual*	1	yes	no
%S31 FASTACT	activation/deactivation of the fast task	see chapter "System Bits" in the *Unity Pro Program Languages and Structure Reference Manual*	0	yes	no
%S32 %S33 %S34 %S35	activation/deactivation of the auxiliary tasks 0-3	see chapter "System Bits" in the *Unity Pro Program Languages and Structure Reference Manual*	0	yes	no
%S38 ACTIVEVT	enabling/inhibition of events	see chapter "System Bits" in the *Unity Pro Program Languages and Structure Reference Manual*	1	yes	no
%S39 EVTOVR	saturation in event processing	see chapter "System Bits" in the *Unity Pro Program Languages and Structure Reference Manual*	0	yes	no
%S50 RTCWRITE	updating of time and date via words %SW50 to %SW53	Normally set to 0, this bit is set to 1 by the program or the terminal: ● set to 0: update of system words %SW50 to %SW53 by the date and time supplied by the PLC real-time clock, ● set to 1: system words %SW50 to %SW53 are no longer updated, therefore making it possible to modify them. ● The switch from 1 to 0 updates the real-time clock with the values entered in words %SW50 to %SW53.	0	yes	yes
%S51 RTCERR	time loss in real-time clock	This system-managed bit set to 1 indicates that the real-time clock is missing or that its system words (%SW50 to %SW53) are meaningless. If set to 1, the clock must be reset to the correct time.	-	no	yes

Description of the System Bits %S59 to %S122

Detailed Description

NOTE: Not all of the system bits can be used in the Quantum Safety PLC. The unusable system bits are marked in the **Quant. Safety** column with no.

The following table gives a description of the system bits %S59 to %S122:

Bit Symbol	Function	Description	Initial State	Write Access	Quant. Safety
%S59 `RTCTUNING`	incremental update of the time and date via word %SW59	Normally set to 0, this bit can be set to 1 or 0 by the program or the terminal: ● set to 0: the system does not manage the system word %SW59, ● set to 1: the system manages edges on word %SW59 to adjust the date and current time (by increment).	0	yes	yes
%S67 `PCMCIABAT0`	state of the application memory card battery	This bit is used to monitor the status of the main battery when the memory card is in the upper PCMCIA slot (all the Atriums, Premiums, and on the Quantums): ● set to 1: main voltage battery is low (application is preserved but you must replace the battery following the so-called predictive maintenance procedure), ● set to 0: main battery voltage is sufficient (application is preserved). Bit %S67 is managed: ● on the PV06 small and medium capacity RAM memory cards (product version written on the card label), i.e. offering memory size under Unity =#768K: TSX MRP P 128K, TSX MRP P 224K TSX MCP C 224K, MCP C 512K, TSX MRP P 384K, TSX MRP C 448K, TSX MRP C 768K, ● under Unity whose version is ≥ 2.02.	-	no	yes
%S68 `PLCBAT`	state of the processor battery	This bit is used to check the operating state of the backup battery for saving data and the program in RAM: ● set to 0: battery present and operational, ● set to 1: battery missing or non-operational.	–	no	yes
%S75 `PCMCIABAT1`	state of the data storage memory card battery	This bit is used to monitor the status of the main battery when the memory card is in the lower PCMCIA slot, see chapter "System Bits" in the *Unity Pro Program Languages and Structure Reference Manual.* **Note:** Data stored on a memory card in slot B are not processed in SIL3 projects.	–	no	no

Bit **Symbol**	Function	Description	Initial State	Write Access	Quant. Safety
%S76 DIAGBUFFCONF	configured diagnostics buffer	This bit is set to 1 by the system when the diagnostics option has been configured. Then, a diagnostics buffer for storage of errors found by diagnostics DFBs is reserved. This bit is read-only.	0	no	yes
%S77 DIAGBUFFFULL	full diagnostics buffer	This bit is set to 1 by the system when the buffer that receives errors from the diagnostics function blocks is full. This bit is read-only.	0	no	yes
%S78 HALTIFERROR	stop in the event of error	Normally at 0, this bit can be set to 1 by the user, to program a PLC stop on application fault: %S15, %S18, %20.	0	yes	yes
%S80 RSTMSGCNT	reset message counters	Normally set to 0, this bit can be set to 1 by the user to reset the message counters %SW80 to %SW86.	0	yes	yes
%S94 SAVECURRVAL	saving adjustment values	see chapter "System Bits" in the *Unity Pro Program Languages and Structure Reference Manual*	0	yes	no
%S118 REMIOERR	General Remote I/O fault	Normally set to 1, this bit is set to 0 by the system when a fault occurs on a device connected to the RIO (Fipio for Premium or Drop S908 for Quantum) remote input/output bus. This bit is reset to 1 by the system when the fault disappears. This bit is not updated if an error occurs on the other buses (DIO, ProfiBus, ASI).	–	no	yes
%S119 LOCIOERR	General inrack I/O fault	Normally set to 1, this bit is set to 0 by the system when a fault occurs on an I/O module placed in 1 of the racks. This bit is reset to 1 by the system when the fault disappears.	–	no	yes
%S120 **%S121** **%S122**	DIO bus faults	see chapter "System Bits" in the *Unity Pro Program Languages and Structure Reference Manual*	–	no	no

Section B.2
System Words

Introduction

This section describes the system words of the Quantum Safety PLC.

For your convenience, all system words of standard Quantum PLCs are listed but only explained further if used in the Quantum Safety PLC.

What Is in This Section?

This section contains the following topics:

Description of the System Words %SW0 to %SW21

Detailed Description

NOTE: Not all of the system words can be used in the Quantum Safety PLC. The unusable system words are marked in the **Quant. Safety** column with no.

The following table gives a description of the system words %SW0 to %SW21:

Word **Symbol**	Function	Description	Initial State	Write Access	Quant. Safety
%SW0 MASTPERIOD	master task scanning period	see chapter "System Objects" in the *Unity Pro Program Languages and Structure Reference Manual*	0	yes	no
%SW1 FASTPERIOD	fast task scanning period	see chapter "System Objects" in the *Unity Pro Program Languages and Structure Reference Manual*	0	yes	no
%SW2, **%SW3**, **%SW4**, **%SW5**	auxiliary task scanning period	see chapter "System Objects" in the *Unity Pro Program Languages and Structure Reference Manual*	0	yes	no
%SW8 TSKINHIBIN	acquisition of task input monitoring	see chapter "System Objects" in the *Unity Pro Program Languages and Structure Reference Manual*	0	yes	no
%SW9 TSKINHIBOUT	monitoring of task output update	see chapter "System Objects" in the *Unity Pro Program Languages and Structure Reference Manual*	0	yes	no
%SW10 TSKINIT	first cycle after cold start	see chapter "System Objects" in the *Unity Pro Program Languages and Structure Reference Manual*	0	no	no
%SW11 WDGVALUE	watchdog duration	Reads the duration of the watchdog. The duration is expressed in milliseconds (20...990 ms). This word cannot be modified.	-	no	yes
%SW12 APMODE	mode of application processor	This word indicates the operating mode of the application processor. Possible values are: ● 16#A501: application processor is in Maintenance Mode. ● 16#5AFE: application processor is in Safety Mode. Any other value is interpreted as an error. This system word is not available for the standard Quantum CPU.	16#A501	no	yes

Word Symbol	Function	Description	Initial State	Write Access	Quant. Safety
%SW13 INTELMODE	mode of Intel processor	This word indicates the operating mode of the Intel Pentium processor. Possible values are: ● 16#501A: application processor is in Maintenance Mode. ● 16#5AFE: application processor is in Safety Mode. Any other value is interpreted as an error. This system word is not available for the standard Quantum CPU.	16#501A	no	yes
%SW14 OSCOMMVERS	commercial version of PLC processor	This word contains the commercial version of the PLC processor. **Example:** 16#0135 version: 01; issue number: 35	-	no	yes
%SW15 OSCOMMPATCH	PLC processor patch version	This word contains the commercial version of the PLC processor patch. It is coded onto the least significant byte of the word. coding: 0 = no patch, 1 = A, 2 = B... **Example:** 16#0003 corresponds to patch C.	-	no	yes
%SW16 OSINTVERS	firmware version number	This word contains the Firmware version number in hexadecimal of the PLC processor firmware. **Example:** 16#0017 version: 2.1; VN: 17	-	no	yes
%SW17 FLOATSTAT	error status on floating operation	see chapter "System Objects" in the *Unity Pro Program Languages and Structure Reference Manual* %SW17.1: Flag not managed by Quantum Safety.	0	yes	yes
%SW18 %SW19 100MSCOUNTER	absolute time counter	%SW18 is the low and %SW19 the high word for calculating durations. Both are incremented every 1/10th of a second by the system (even when the PLC is in STOP, they are no longer incremented if it is powered down). They can be read and written by the user program or by the terminal.	0	yes	yes
%SW20 %SW21 MSCOUNTER	absolute time counter	The low word %SW20 and the high word %SW21 are incremented every 1/1000th of a second by the system (even when the PLC is in STOP, they are no longer incremented if it is powered down). They can be read by the user program or by the terminal. %SW20 and %SW21 are reset on a cold start, but not on a warm start.	0	no	yes

Description of the System Words %SW30 to %SW59

Detailed Description

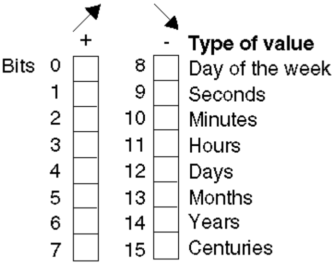
NOTE: Not all of the system words can be used in the Quantum Safety PLC. The unusable system words are marked in the **Quant. Safety** column with no.

The following table gives a description of the system words %SW30 to %SW59:

Word `Symbol`	Function	Description	Initial State	Write Access	Quant. Safety
%SW30 `MASTCURRTIME`	master task execution time	This word indicates the execution time of the last master task cycle (in ms).	-	no	no
%SW31 `MASTMAXTIME`	maximum master task execution time	This word indicates the longest master task execution time since the last cold start (in ms).	-	no	yes
%SW32 `MASTMINTIME`	minimum master task execution time	This word indicates the shortest master task execution time since the last cold start (in ms).	-	no	yes
%SW33 to %SW35	fast task execution times	see chapter "System Objects" in the *Unity Pro Program Languages and Structure Reference Manual*	-	no	no
%SW36	NTP number of seconds (LSB)	This word indicates the number of seconds passed since January 1st, 1980 at 00:00 (LSB part). It reflects the NTP time coming from the 140 NOE 771 11 module. This word is refreshed internally between two NTP synchronizations.	0	no	yes
%SW37	NTP number of seconds (MSB)	This word indicates the number of seconds passed since January 1st, 1980 at 00:00 (MSB part). It reflects the NTP time coming from the 140 NOE 771 11 module. This word is refreshed internally between two NTP synchronizations.	0	no	yes
%SW38	NTP number of milliseconds	This word indicates the number of milliseconds added to the NTP number of seconds (%SW36 and %SW37). It reflects the NTP time coming from the 140 NOE 771 11 module. This word is refreshed internally between two NTP synchronizations.	0	no	yes

Word Symbol	Function	Description	Initial State	Write Access	Quant. Safety
%SW39	status of the NTP timestamps in ms	Meaning of the different bits of %SW39 word: ● %SW39.0 (managed by the controller): ❍ =0, the time value is not available or the time has not been updated within last 2 minutes ❍ =1, the time value is available or the time has been updated within last 2 minutes ● %SW39.1 (managed by the 140 NOE 771 11 status): ❍ =0, the NTP server time value is not available ❍ =1, the updated time value is received from the NTP server and has been sent to the module (at least once) ● %SW39.2 (managed by the CPU): ❍ =0, the time value in %SW36 to %SW38 words differs from the last NTP server time received by more than 2 seconds. The last NTP server time received has been ignored. ❍ =1, the time value in %SW36 to %SW38 words are consistent with the last NTP server time received (less than 2 seconds difference). The time value in %SW36 to %SW38 words is filtered with a slope of 1ms/s to reach the last NTP server time received. ● %SW39.3 to %SW39.7: not used ● %SW39.8 (control that can be set by the application): ❍ =0, no action ❍ =1. When set to 1, the CPU will accept the next NTP server time received without filtering (1 ms/s) and without consistency check (difference between time value in %SW36 to %SW38 words and NTP server time). After the next NTP server time is received, the %SW39.8 bit is automatically reset to 0 by the controller. ● %SW39.9 to %SW39.15: not used	0	yes	yes
%SW40 to %SW47	auxiliary tasks execution times	see chapter "System Objects" in the *Unity Pro Program Languages and Structure Reference Manual*	-	no	no

Word **Symbol**	Function	Description	Initial State	Write Access	Quant. Safety
%SW48 IOEVTNB	number of events	see chapter "System Objects" in the *Unity Pro Program Languages and Structure Reference Manual*	0	yes	no
%SW49 DAYOFWEEK %SW50 SEC %SW51 HOURMIN %SW52 MONTHDAY %SW53 YEAR	real-time clock function	System words containing date and current time (in BCD): ● %SW49: day of the week: ❍ 1 = Monday, ❍ 2 = Tuesday, ❍ 3 = Wednesday, ❍ 4 = Thursday, ❍ 5 = Friday, ❍ 6 = Saturday, ❍ 7 = Sunday, ● %SW50: Seconds (16#SS00), ● %SW51: Hours and Minutes (16#HHMM), ● %SW52: Month and Day (16#MMDD), ● %SW53: Year (16#YYYY). These words are managed by the system when the bit %S50 is set to 0. These words can be written by the user program or by the terminal when the bit %S50 is set to 1.	-	yes	yes
%SW54 STOPSEC %SW55 STOPHM %SW56 STOPMD %SW57 STOPYEAR %SW58 STOPDAY	real-time clock function on last stop	System words containing date and time of the last power outage or PLC stop (in Binary Coded Decimal): ● %SW54: Seconds (00SS), ● %SW55: Hours and Minutes (HHMM), ● %SW56: Month and Day (MMDD), ● %SW57: Year (YYYY), ● %SW58: the most significant byte contains the day of the week (1 for Monday through to 7 for Sunday), and the least significant byte contains the code for the last stop: ❍ 1 = change from RUN to STOP by the terminal or the dedicated input, ❍ 2 = stop by watchdog (PLC task or SFC overrun), ❍ 4 = power outage or memory card lock operation, ❍ 5 = stop on hardware fault, ❍ 6 = stop on software fault. Details on the type of software fault are stored in %SW125.	-	no	yes

Word **Symbol**	Function	Description	Initial State	Write Access	Quant. Safety
%SW59 ADJDATETIME	adjustment of current date	Contains 2 8-bit series to adjust the current date. The action is performed on the rising edge of the bit. This word is enabled by bit %S59=1. In the following illustration, bits in the left column increment the value, and bits in the right column decrement the value:  + - **Type of value** Bits 0 8 Day of the week 1 9 Seconds 2 10 Minutes 3 11 Hours 4 12 Days 5 13 Months 6 14 Years 7 15 Centuries	0	yes	yes

Description of the System Words %SW60 to %SW127

Detailed Description

Not all of the system words can be used in the Quantum Safety PLC. In the following table the unusable system words are marked **no** in the **Quant. Safety** column.

This table gives a description of the system words %SW60 to %SW127:

Word **Symbol**	Function	Description	Initial State	Write Access	Quant. Safety
%SW60 HSB_CMD	Quantum Hot Standby command register	Meaning of the different bits of the word %SW60: ● %SW60.0 = 1 invalidates the commands entered in the display (keypad). ● %SW60.1 ❍ = 0 sets PLC A to OFFLINE mode. ❍ = 1 sets PLC A to RUN mode. ● %SW60.2 ❍ = 0 sets PLC B to OFFLINE mode. ❍ = 1 sets PLC B to RUN mode. ● %SW60.3 ❍ = 0 If an application mismatch is detected, standby PLC is forced to OFFLINE mode. ❍ = 1 Standby PLC operates normally even if a mismatch occurs. ● %SW60.4 ❍ = 0 authorizes an update of the firmware only after the application has stopped. ❍ = 1 authorizes an update of the firmware without the application stopping. ● %SW60.5= 1 application transfer request from the standby to the primary. ● %SW60.8 ❍ = 0 address is switched on Modbus port 1 during a primary swap. ❍ = 1 address is not switched on in Modbus port 1 during a primary swap.	0	yes	no

Word Symbol	Function	Description	Initial State	Write Access	Quant. Safety
%SW61 HSB_STS	Quantum status register	Meaning of the different bits of the word %SW61: ● %SW61.0 and %SW61.1 PLC operating mode bits: ○ %SW61.1 = 0, %SW61.0=1: OFFLINE mode ○ %SW61.1 = 1, %SW61.0=0: primary mode ○ %SW61.1 = 1, %SW61.0=1: secondary mode (Standby) ● %SW61.2 and %SW61.3 operating mode bits from the other PLC ○ %SW61.3 =0, %SW61.2=1: OFFLINE mode ○ %SW61.3 = 1, %SW61.2=0: primary mode ○ %SW61.3 = 1, %SW61.2=1: secondary mode (Standby) ○ %SW61.3 = 0, %SW61.2=0: the remote PLC is not accessible (switched off, no communication) ● %SW61.4 ○ = 0 the applications on both PLCs are identical ○ = 1 the applications on both PLCs are not identical ● %SW61.5 ○ = 0 the PLC is used as unit A ○ = 1 the PLC is used as unit B ● %SW61.7 ○ = 0 same PLC OS version ○ = 1 different PLC OS version ● %SW61.8 ○ = 0 same copro OS version ○ = 1 different copro OS version ● %SW61.12 ○ = 0 information given by bit 13 is not relevant ○ = 1 information given by bit 13 is valid ● %SW61.13 ○ = 0 NOE address set to IP. ○ = 1 NOE address set to IP + 1. ● %SW61.15 ○ = 0 Hot Standby not activated ○ = 1 Hot Standby activated	0	no	yes

Word `Symbol`	Function	Description	Initial State	Write Access	Quant. Safety

%SW62
`HSBY_REVERSE0`
%SW63
`HSBY_REVERSE1`
	Transfer word	These 2 words may be added to the first section of the master task. They are then transferred automatically from the standby processor to update the primary PLC.
		They may be read on the primary PLC and be used as primary application parameters.
			0	yes	yes

%SW70 `WEEKOFYEAR`	real-time clock function	System word containing the number of the week in the year: 1 to 52.	–		yes
%SW71 `KEY_SWITCH`	position of the switches on the Quantum front panel	This word provides the image of the positions of the switches on the front panel of the Quantum processor. This word is updated automatically by the system. ● `%SW71.0` = 1 switch in the "Memory protected" position ● `%SW71.1` = 1 switch in the "STOP" position ● `%SW71.2` = 1 switch in the "START" position ● `%SW71.8` = 1 switch in the "MEM" position ● `%SW71.9` = 1 switch in the "ASCII" position ● `%SW71.10` = 1 switch in the "RTU" position ● `%SW71.3` to 7 and 11 to 15 are not used	0	no	yes
%SW75 `TIMEREVTNB`	timer-type event counter.	See chapter "System Objects" in the *Unity Pro Program Languages and Structure Reference manual*.	0		no
%SW76 `DLASTREG`	diagnostics function: recording	Result of the last registration: ● = 0 if the recording was successful ● = 1 if the diagnostics buffer has not been configured ● = 2 if the diagnostics buffer is full	0		yes
%SW77 `DLASTDEREG`	diagnostics function: non-recording	Result of the last de-registration: ● = 0 if the non-recording was successful ● = 1 if the diagnostics buffer has not been configured ● = 21 if the error identifier is invalid ● = 22 if the error has not been recorded	0		yes
%SW78 `DNBERRBUF`	diagnostics function: number of errors	Number of errors currently in the diagnostics buffer.	0		yes

Word **Symbol**	Function	Description	Initial State	Write Access	Quant. Safety

%SW80
`MSGCNT0`
%SW81
`MSCNT1`
	message management	These words are updated by the system, and can also be reset using `%S80`.
		- `%SW80`: Number of Modbus messages sent by the system as client on all communication ports except USB and Ethernet copro. **NOTE:** Modbus messages sent by the system as Master are not counted in this word.
		- `%SW81`: Number of Modbus messages received by the system as client on all communication ports except USB and Ethernet copro. **NOTE:** Modbus messages received as response to the requests sent by the system, as Master, are not counted in this word.
			0	yes	yes

%SW87 `MSTSERVCNT`	communication flow management	Number of requests processed by synchronous server per master (MAST) task cycle.	0		yes
%SW90 `MAXREQNB`	maximum number of requests processed per master task cycle	This word is used to set a maximum number of requests which can be processed by the PLC per master task cycle. When the CPU is the server: This number of requests must be between 2 (minimum) and N+4 (maximum). N: Number differs depending on the model. When the CPU is the client: N: Number differs depending on the model. The value 0 does not work. If a value is entered that is outside of the range, the value N that is taken into account. See also chapter "System Objects" in the *Unity Pro Program Languages and Structure Reference Manual*.	0	yes	yes
%SW108 `FORCEDIOIM`	number of forced I/O module bits	This system word counts the number of forced I/O module bits. This word is incremented for every forcing, and decremented for every un-forcing.	0	no	yes
%SW110	number of unrestricted memories area for %M	This system word gives information on the size of the unrestricted memory area for %M. This system word is not available for the standard Quantum CPU.	–	no	yes
%SW111	number of unrestricted memories area for %MW	This system word gives information on the size of the unrestricted memory area for %MW. This system word is not available for the standard Quantum CPU.	–	no	yes

Word **Symbol**	Function	Description	Initial State	Write Access	Quant. Safety
%SW124 CPUERR	type of system fault	This system word is updated if the PLC is set to error state. The possible values are as follows: ● 0x0065: execution of HALT instruction impossible ● 0x0080: system watchdog If the PLC is set to Safety error state, the content of %SW125 is updated and can be read after the next restart of the PLC (see below).	–	no	yes

Word **Symbol**	Function	Description	Initial State	Write Access	Quant. Safety
%SW125 BLKERRTYPE	last fault detected	The code of the last fault detected is given in this word. If %S78 is set to 1, the following detected errors cause the PLC to stop. %S15, %S18 and %S20 are activated independently of %S78: ● 16#0002: PCMCIA signature not verified ● 16#2258: execution of HALT instruction ● 16#2302: call to a not supported system function in a user function block ● 16#9690: error of application CRC detected in background ● 16#DE87: calculation error on floating-point numbers (%S18, these errors are listed in the word %SW17) ● 16#DEB0: watchdog overflow (%S11) ● 16#DEF1: character string transfer error (%S15) ● 16#DEF2: arithmetic or division by 0 error (%S18) ● 16#DEF3: index overflow (%S20) **Note:** The codes 16#8xxx and 16#7xxx do not stop the application and indicate an error on function blocks. In case of an SIL3 related error, the PLC stops. After power off and restart of the PLC, %SW 125 contains the code of the cause of the error: ● 0x5AF1: sequence check error (unpredictable execution in CPU) ● 0x5AF2: error in memory (corrupt address) ● 0x5AF3: comparison error (execution results of Intel and application processor differ) ● 0x5AF4: real-time clock error ● 0x5AF5: error initializing double code execution ● 0x5AF6: watchdog activation error ● 0x5AF7: error during memory check (takes more than 8 hours) ● 0x5AF8: error in memory check (corrupt RAM) **Note:** %SW125 is only reset after init or complete download or restart (it always contains the last fault detected).	–	no	yes

Word **Symbol**	Function	Description	Initial State	Write Access	Quant. Safety
%SW126 ERRADDR0 **%SW127** ERRADDR1	blocking fault instruction address	Address of the instruction that generated the application blocking fault. For 16-bit processors: ● %SW126 contains the offset for this address ● %SW127 contains the segment number for this address. For 32-bit processors: ● %SW126 contains the least significant word for this address ● %SW127 contains the most significant word for this address The content of %SW126 and %SW127 is for Schneider Electric use only.	0	no	yes

For the description of the system words %SW128 to %SW339 and %SW535 to %SW640, see the chapter "Quantum Specific System Words" in the *Unity Pro Program Languages and Structure Reference Manual*. The system words %SW340 to %SW534 are not used in Quantum Safety PLCs.

Glossary

!

!

NOTE: For terms taken from the IEC 61508 standard, refer to the standard for complete definitions.

1002D diagnostic configuration

X out of Y

For example 1 out of 2. Voting and redundancy capacity of a Safety-Related System.

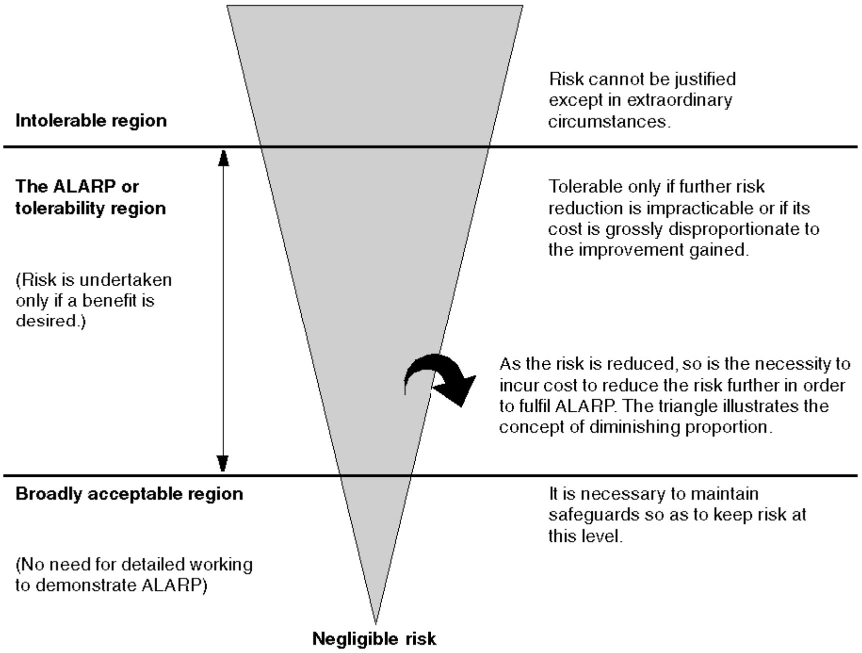
D in 1oo2D refers to diagnostics. Hence, D in 1oo2D means 1 out of 2 with diagnostics.

A

ALARP

as low as is reasonably practicable

(Definition IEC 61508)


Intolerable region
Risk cannot be justified except in extraordinary circumstances.
The ALARP or tolerability region
(Risk is undertaken only if a benefit is desired.)
Tolerable only if further risk reduction is impracticable or if its cost is grossly disproportionate to the improvement gained.
As the risk is reduced, so is the necessity to incur cost to reduce the risk further in order to fulfil ALARP. The triangle illustrates the concept of diminishing proportion.
Broadly acceptable region
(No need for detailed working to demonstrate ALARP)
It is necessary to maintain safeguards so as to keep risk at this level.
Negligible risk

C

CCF

common cause failure

failure, which is the result of 1 or more events, causing coincident failures of 2 or more separate channels in a multiple channel system, leading to system failure

(Definition IEC 61508)

The common cause factor in a dual channel system is the crucial factor for the probability of failure on demand (PFD) for the whole system.

cold start

Cold start refers to starting the computer from power off.

CPU

central processing unit

CRC

cyclic redundancy check

D

DC

diagnostic coverage

fractional decrease in the probability of dangerous hardware failures resulting from the operation of the automatic diagnostic tests

(Definition IEC 61508)

The fraction of the possible dangerous failures λ_D is divided into failures which are detected by diagnostics and failures which remain undetected.

$\lambda_D=\lambda_{DD} + \lambda_{DU}$

The diagnostic coverage (DC) defines the fraction of the dangerous failures which are detected.

$\lambda_{DD} =\lambda_D$. DC

$\lambda DU=\lambda_D (1 - DC)$

The definition may also be represented in terms of the following equation, where DC is the diagnostic coverage, λ_{DD} is the probability of detected dangerous failures and λ_D total is the probability of total dangerous failures:

$$DC= \frac{\sum \lambda_{DD}}{\sum \lambda_{Dtotal}}$$

DDT

derived data type

A derived data type is user defined.

DFB

derived function block

DIO

distributed input/output

DLL

dynamic link library

E

E/E/PES

electrical/electronic/programmable electronic system

(Definition IEC 61508)

System for control, protection or monitoring based on 1 or more electrical/electronic programmable electronic (E/E/PE) devices. This includes elements of the system such as power supplies, sensors and other input devices, data highways and other communication paths, and actuators and other output devices.

EDT

elementary data type

An elementary data type is predefined.

EF

elementary function

EFB

elementary function block

EMC

electromagnetic compatibility

The term refers to the origin, control, and measurement of electromagnetic effects on electronic systems.

EN

European Norm

This is the official European standard.

error

discrepancy between a computed, observed or measured value or condition and the true, specified or theoretically correct value or condition

(Definition IEC 61508)

ESD

emergency shutdown

EUC

equipment under control

(Definition IEC 61508)

This term designates equipment, machinery, apparatuses or plants used for manufacturing, process, transportation, medical or other activities.

F

failure

termination of the ability of a functional unit to perform a required function

(Definition IEC 61508)

fault

abnormal condition that may cause a reduction in, or loss of, the capability of a functional unit to perform a required function

(Definition IEC 61508)

FBD

functional block diagram

This is an IEC 61131-3 programming language for PLC user logic.

FFB

function/function block

FMEA

failure modes and effects analysis

FMECA

failure modes and effects criticality analysis

Functional Safety

part of the overall safety relating to the EUC and the EUC control system which depends on the correct functioning of the E/E/PE safety-related systems, other technology safety-related systems and external risk reduction facilities

(Definition IEC 61508)

A system is defined functionally Safe if random, systematic and common cause failures do not lead to malfunctioning of the system and do not result in injury or death of humans, spills to the environment and loss of equipment or production:
- Functional Safety deals with the part of the overall Safety that depends on the correct functioning of the Safety-Related System.
- Functional Safety applies to products as well as organizations.

H

HALT

high accelerated life tests

HFT

hardware fault tolerance

(Definition IEC 61508)

A hardware fault tolerance of N means that N + 1 faults could cause a loss of the Safety Function, for instance:
- HFT = 0: The 1st failure could cause a loss of the Safety Function
- HFT = 1: 2 faults in combination could cause a loss of the Safety Function. (There are 2 different paths to go to a Safe state. Loss of the Safety Function means that a Safe state cannot be entered.

HMI

human-machine interface

HSBY

Hot Standby

I

IEC

International Electrotechnical Commission

IEC 61508

The IEC 61508 standard is an international standard that addresses Functional Safety of electrical / electronic / programmable electronic Safety-Related Systems. It applies to any kind of Safety-Related System in any industry wherever there are no product standards.

IL

instruction list

This is an IEC 61131-3 programming language for PLC user logic.

L

LCD

liquid crystal display

LD

ladder diagram

This is an IEC 61131-3 programming language for PLC user logic.

M

MTBF

mean time between failures

MTTF

mean time to failure

MTTR

mean time to repair

N

NFPA

National Fire Protection Association

This is a body for establishing codes and standards for fire protection, electrical and machine Safety in the U.S.

non-interfering module

Non-interfering modules are modules that are not directly used to control the Safety Function. They do not interfere with the Safety modules (either during normal operation or if there is a fault).

NTP

Network Time Protocol

P

PELV

protected extra low voltage

PES

programmable electronic system

(Definition IEC 61508)

System for control, protection or monitoring based on 1 or more programmable electronic devices, including elements of the system such as power supplies, sensors and other input devices, data highways and other communication paths, and actuators and other output devices.

PES is another term for a computer control system or PLC.

PFD

probability of failure on demand

(Definition IEC 61508)

For a single channel system the average probability of a failure on demand is calculated as follows:

$$\text{PFD(t)}_{\text{Av}} = \frac{1}{2}\lambda_{DU} \bullet t$$

For a dual channel system the average probability of a failure on demand is calculated as follows:

$$\text{PFD(t)}_{\text{Av}} = \lambda_{DUCH1} \bullet \lambda_{DUCH2} \bullet t^2 + CC$$

For a dual channel system, also the Common Cause effect (CC) must be considered. The common cause effect ranges from 1% to 10% of PFD_{CH1} and PFD_{CH2}. (=1/RRF).

PFH

probability of failure per hour

(Definition IEC 61508)

PLC

programmable logic controller

project

A project is a user application in Unity Pro XLS.

proof test

periodic test performed to detect failures in a safety-related system so that, if necessary, the system can be restored to an "as new" condition or as close as practical to this condition

(Definition IEC 61508)

proof test interval

The proof test interval is the time period between proof tests.

PRT

PLC reaction time

The PLC Reaction Time is the time which passes between a signal is detected at the input module terminal and the reaction is set at the output module terminal.

PS

power supply

PST

process safety time

The process safety time is defined as the period of time between a failure occurring in EUC or the EUC control system (with the potential to give rise to a hazardous event) and the occurrence of the hazardous event if the safety function is not performed.

(Definition IEC 61508)

Q

QSE

environment system qualification

R

RAM

random access memory

random hardware failure

failure, occurring at a random time, which results from 1 or more of the possible degradation mechanisms in the hardware

(Definition IEC 61508)

RIO

remote input/output

risk

combination of the probability of occurrence of harm and the severity of that harm

(Definition IEC 61508)

Risk is calculated using the following equation: R=S*H

The letters stand for:

Letter	Meaning
R	risk
S	extent of the damage
H	frequency of occurrence of the damage

RM

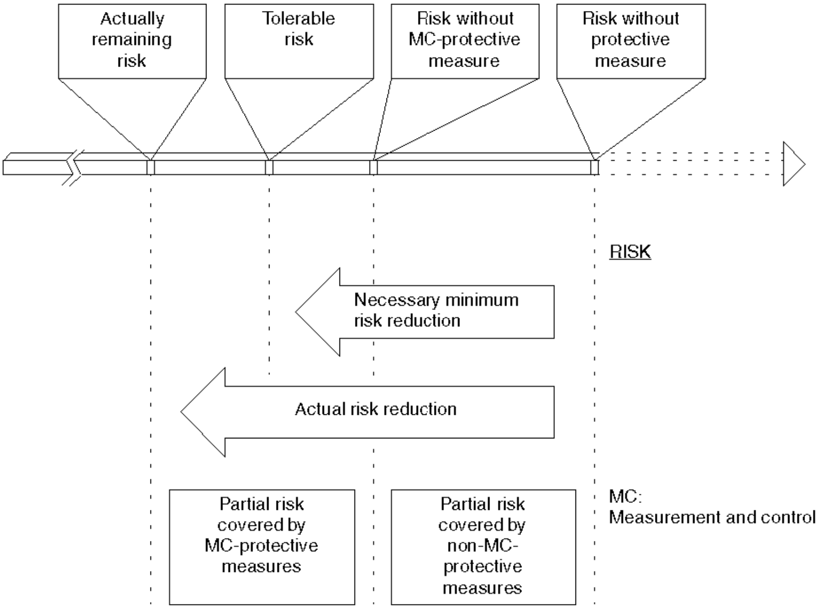
requirements management

RRF

risk reduction factor

(Definition IEC 61508)

The risk reduction factor equals 1/PFD.


Actually remaining risk
Tolerable risk
Risk without MC-protective measure
Risk without protective measure
RISK
Necessary minimum risk reduction
Actual risk reduction
Partial risk covered by MC-protective measures
Partial risk covered by non-MC-protective measures
MC:
Measurement and control

RTC

real-time clock

S

Safety Function

function to be implemented by an E/E/PE safety-related system, other technology safety-related system or external risk reduction facilities, which is intended to achieve or maintain a safe state for the EUC, in respect of a specific hazardous event

(Definition IEC 61508)

Safety Integrity

probability of a safety-related system satisfactorily performing the required safety functions under all the stated conditions within a stated period of time

(Definition IEC 61508)

Safety PLC

Quantum Safety PLC (140 CPU 651 60S or 140 CPU 671 60S)

Safety variable

variable used to implement a Safety Function in a Safety-Related System

Safety-Related System

This term designates a system that both

- implements the required Safety Functions necessary to achieve or maintain a Safe state for the EUC and
- is intended to achieve, on its own or using other E/E/PE Safety-Related Systems, other technology Safety-Related Systems, or external risk reduction facilities, the necessary Safety Integrity for the required Safety Functions.

SFC

sequential function chart

This is an IEC 61131-3 programming language for PLC user logic.

SFF

safe failure fraction

SFR

Safety Functional requirement

Safety Functional requirements are derived from the hazard analysis and define what the function does, for instance the Safety Function to be performed.

SIL

NOTE: For complete definitions and parameters related to SIL ratings refer to IEC 61508, "Functional safety of electrical/electronic/programmable electronic safety related systems". Provided here is a partial definition.

safety integrity level

discrete level (1 out of a possible 4) for specifying the safety integrity requirements of the safety functions to be allocated to the E/E/PE safety-related systems, where safety integrity level 4 has the highest level of safety integrity and safety integrity level 1 has the lowest

(Definition IEC 61508)

SIL2 project (application)

A project (application) that uses a Quantum Safety PLC (140 CPU 651 60S V1.00 or 140 CPU 671 60S V1.00) to implement Safety Functions in a Safety-Related System.

SIL3 project (application)

A project (application) that uses a Quantum Safety PLC (140 CPU 651 60S V2.00 or 140 CPU 671 60S V2.00) to implement Safety Functions in a Safety-Related System.

SIR

Safety Integrity requirement

Safety Integrity requirements are derived from a risk assessment and describe the likelihood of a Safety Function to be performed satisfactorily, for instance the degree of certainty necessary for the Safety Function to be carried out.

sniffing

reading the configuration out of a PLC

SRS

safety requirements specification

specification containing all the requirements of the safety functions that have to be performed by the safety-related systems

(Definition IEC 61508)

SSC

system Safety concept

This is a detailed description of the system architecture, configuration and diagnostics required to achieve Functional Safety.

ST

structured text

This is an IEC 61131-3 programming language for PLC user logic.

Statement of Consequence

This is the last line within all special messages. It begins with "**Failure to follow these instructions**..."

systematic failure

failure related in a deterministic way to a certain cause, which can only be eliminated by a modification of the design or of the manufacturing process, operational procedures, documentation or other relevant factors

(Definition IEC 61508)

T

TÜV

Technischer Überwachungsverein

(German for Association for Technical Inspection)

U

UMA

unrestricted memory area

It is a specially dedicated memory area for bits and words which is not write protected.

V

VDE

Verband Deutscher Elektroingenieure

This is the German equivalent of the IEEE.

W

warm start

Warm start refers to restarting the computer without turning the power off.

Index